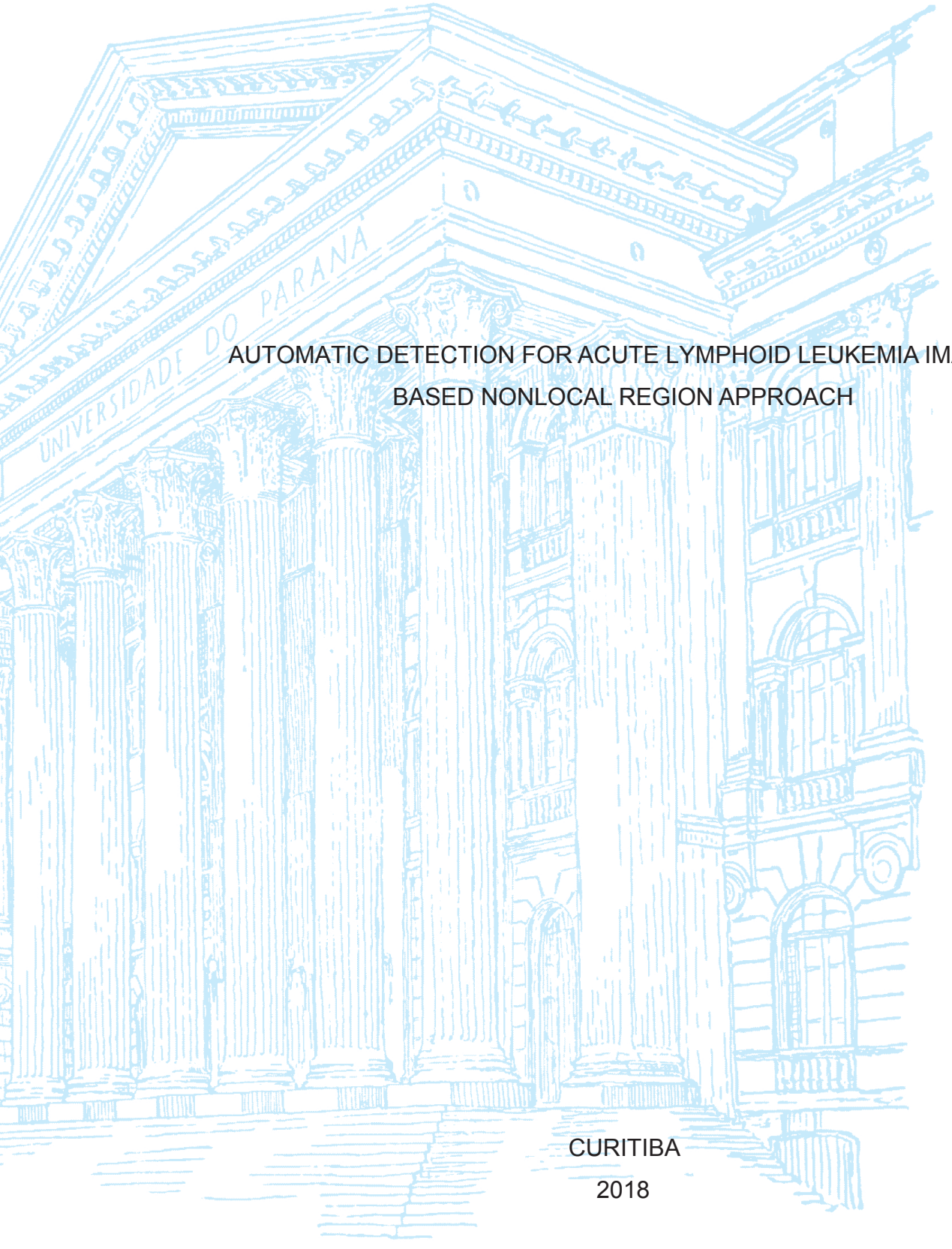
UNIVERSIDADE FEDERAL DO PARANÁ

JOÃO FELIPE LOPES DE SUS



AUTOMATIC DETECTION FOR ACUTE LYMPHOID LEUKEMIA IMAGES BASED NONLOCAL REGION APPROACH

CURITIBA

2018

JOÃO FELIPE LOPES DE SUS

AUTOMATIC DETECTION FOR ACUTE LYMPHOID LEUKEMIA IMAGES BASED ON LOCAL REGION APPROACH

Dissertação apresentada ao Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Informática.
Orientador: Prof. Dr. Lucas Ferrari de Oliveira

CURITIBA

2018

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

S964a Sus, João Felipe Lopes de
Automatic detection for acute lymphoid leukemia images based on local region approach / João Felipe Lopes de Sus. – Curitiba, 2018.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2018.

Orientador: Lucas Ferrari de Oliveira .

1. Leucemia. 2. Processamento de imagens - Técnicas digitais. 3. Sistemas de reconhecimento de padrões. I. Universidade Federal do Paraná. II. Oliveira, Lucas Ferrari de. III. Título.

CDD: 616.15

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

MINISTÉRIO DA EDUCAÇÃO
SETOR SETOR DE CIENCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **JOÃO FELIPE LOPES DE SUS** intitulada: **Automatic Detection for Acute Lymphoid Leukemia Images Based on Local Region Approach**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 03 de Setembro de 2018.

LUCAS FERRARI DE OLIVEIRA
Presidente da Banca Examinadora (UFPR)

CAROLINA CARDOSO DE MELLO PRANDO
Avaliador Externo (IPPP)

LUIZ EDUARDO SOARES DE OLIVEIRA
Avaliador Interno (UFPR)

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Ciências Exatas
Coordenação de Pós-Graduação em Informática

RESUMO

A leucemia linfoide aguda (LLA) é o tipo de câncer mais comum a se manifestar na infância, apesar de apresentar rápida evolução em seu quadro clínico a LLA possui relativamente baixa mortalidade quando identificada e tratada em seu estágio inicial. Como o diagnóstico de LLA é feito por médicos hematologistas com base na análise microscópica de lâminas contendo amostras de sangue periférico, o que pode ser considerado um trabalho lento e cansativo impactando no desempenho do médico, o desenvolvimento de ferramentas que auxiliem neste processo é uma necessidade real. A proposta deste trabalho é apresentar um algoritmo capaz de segmentar os leucócitos existentes, extrair e selecionar características gerando uma representação compacta e por fim utilizar um classificador para diferenciar imagens de sangue periférico de pacientes saudáveis de pacientes portadores de LLA. A base de imagens ALL_IDB foi escolhida para ser utilizada por ser uma base de domínio público e também utilizada em outros trabalhos permitindo comparações precisas, e apresentar diversas dificuldades encontradas no trabalho com imagens provenientes de microscópio, como diferentes tipos de iluminação e zoom. Das 108 imagens utilizadas nos testes 107 foram classificadas corretamente, resultando em uma acurácia de 0,99 sendo este valor maior que o melhor trabalho encontrado na literatura atual, mesmo o único caso classificado erroneamente foi um falso positivo o que no contexto da aplicação é menos grave do que um falso negativo.

Palavras-chave: Leucemia Linfoide Aguda, Processamento de Imagens, Reconhecimento de Padrões, Texturas, Classificadores Lineares

ABSTRACT

Acute lymphoblastic leukemia (ALL) is the most common type of cancer in childhood. Despite presenting a rapid evolution in its clinical condition, ALL has a relatively low mortality when identified and treated in its initial stage. Due to the fact that the ALL diagnosis is made by hematologists based on the microscopic analysis of the peripheral blood smear slices, which can be considered a tedious and tiring work, impacting on the doctor's performance, the development of tools that would help in this process is a real necessity. Thus, the purpose of this work is to present an algorithm capable of segmenting the existing leukocytes from blood smear images, extracting and selecting the most representative features, generating a compact representation, so as to finally use a classifier to differentiate the peripheral blood smear images of healthy patients from patients with ALL. The ALL_IDB image base was chosen for being a public domain base and also used in other works, thus allowing accurate comparisons, as well as revealing several difficulties that are faced when working with microscopic images, such as different types of lighting and distinct zoom levels. The final results were expressive and reached an accuracy of 0.99, where, from the 108 images used in the tests, 107 were classified correctly. This result is higher than the best one found in the latest literature, and the only image classified as being wrong was a false positive which in the application context is not the worse case scenario.

Keywords: Acute Lymphoid Leukemia, Image Processing, Pattern Recognition, Textures, Linear Classifiers

LISTA DE ILUSTRAÇÕES

FIGURE 1 – Erythrocytes in a blood smear, where a nucleus absence and the disco shape can be noticed. **Source**: (BARRETO et al., 2014) . . . 22

FIGURE 2 – Leukocyte classes, (a) neutrophil, (b) eosinophil, (c) basophil, (d) monocyte, (e) leukocyte. **Source**: the author 22

FIGURE 3 – (a) original grayscale image, (b) result of the sample threshold application using T as 127, (c) result of Otsu's method. It is observed that image (c) preserved many more details of the original image than image (b). **Source**: the author 25

FIGURE 4 – The RGB model interpreted as a three-dimensional cube. **Source**: <https://upload.wikimedia.org/wikipedia/commons/a/af/RGB_color_solid_cube.png> 25

FIGURE 5 – Geometric representation of the HSV model. **Source**: <https://software.intel.com/en-us/node/503873> 26

FIGURE 6 – L*a*b color model representation, illustrate the opposite color theory. **Source**: (LIU et al., 2014) 27

FIGURE 7 – Structuring element examples, with distinct shapes. **Source:** <https://sofaltatestar.wordpress.com/2010/01/06/morfologia-matematica-para-imagens-em-tons-de-cinza/> 27

FIGURE 8 – Illustrates the result of an erosion and dilation morphological operation application. (a) original image, (b) erosion result, (c) dilatation result. **Source**: <https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html> 28

FIGURE 9 – Displays the results of opening and closing operations on a noised image, where (a) is the result of an opening used to remove small objects, and (b) is the result of a closing operation used to close all small holes on the object. **Source**: <https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html> 29

FIGURE 10 – (a) Binary image composed only of contours; (b) Result of region growth, applied in image (a), using at least one external point and another internal one for the contour as seed points. The contours have their value inverted for illustrative purposes. **Source**: (FILHO; NETO, 1999) 30

FIGURE 11 – Shows result variation of the K-means algorithm application with distinct values attributed to *K*. **Source**: <http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_ml/py_kmeans/py_kmeans_opencv/py_kmeans_opencv.html> 31
FIGURE 12 – Displays the phases of watershed algorithm applied on image (a); (b) is the topographic representation; images (c) to (g) contain the results on several floods; (h) contains the final watershed contours. **Source**: (GONZALEZ; WOODS, 2010) 31
FIGURE 13 – Local Binary Pattern calculation example. **Source**: (GORODNICHY et al., 2014) 33
FIGURE 14 – Architecture of the CNN LeNet-5. **Source**: (LECUN et al., 1998) . 34
FIGURE 15 – Residual block architecture. **Source**: (DESHPANDE, 2016) 34
FIGURE 16 – It displays a boundary that splits dots of different classes, according to its color in two different scenarios, where there is a linear boundary on A and a nonlinear boundary on B. **Source**: <https://i.stack.imgur.com/OrcTJ.png> 36
FIGURE 17 – Displays the result of the *LDA* data transformation by rotating it. **Source**: <http://www.tutorial.freehost7.com/human_face_recognition/linear_discriminant_analysis.htm> 37
FIGURE 18 – Displays Perceptron architecture. **Source**: <http://abhay.harpale.net/blog/machine-learning/a-hands-on-tutorial-on-the-perceptron-learning-algorithm/> 38
FIGURE 19 – *Multilayer Perceptron* with only one hidden layer. **Source**: <http://scikit-learn.org/stable/modules/neural_networks_supervised.html> 39
FIGURE 20 – Optimal decision boundary found by SVM through a hyperplane based on the margins of each class. **Source**: <http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html> 40
FIGURE 21 – Modules of an automatic system for acute leukemia detection, the paper proposed implements only the gray box phase. **Source**: (SCOTTI, 2006). 42
FIGURE 22 – Architecture of Neural Network Model. **Source**: (WU et al., 2009). . 44
FIGURE 23 – Flowchart of the proposed dual-threshold segmentation scheme. **Source**: (LI et al., 2016). 47
FIGURE 24 – The overview of the proposed method. **Source**: (LE et al., 2015). . . 49
FIGURE 25 – (a) ROIs used in training phase, blue boxes refers to leukocytes, red erythrocytes and green to background, (b) mask resulting of SVM pixels classification. **Source**: (LODDO et al., 2016). 50

FIGURE 26 – Diagram of the proposed method from blood image to the ALL classification via identification of white blood cells (WBCs). **Source:** (PUTZU et al., 2014). 55
FIGURE 27 – Images that show many different kinds of lighting and the zoom level present on the ALL_IDB1 base. **Source**: (LABATI et al., 2011) . . 60
FIGURE 28 – ALL_IDB2 image examples that show the existence of erythrocytes and incomplete leukocytes on the extremities, and one leukocyte in the center. **Source**: (LABATI et al., 2011) 60
FIGURE 29 – Complete pipeline proposed, where each colorful dash represents one step described in the previous section, where the green dashes represent the first step, the blue ones the second step, the orange ones the third step, the yellow ones the fourth step and the purple ones the fifth step. **Source**: the author. 61
FIGURE 30 – This figure exemplifies the steps used in leukocyte extraction. Where (a) is the ALL_IDB1 image, (b) is the result of the pipeline that segmented all leukocytes existing on (a), (c) calculates the center of mass of each leukocyte, represented by a green dot, (d) represents the cutout images resulting from this pipeline. **Source**: the author. . 62
FIGURE 31 – Displays the result of each step of the algorithm 3, (a) original image, (b) L channel, (c) L blurred, (d) inverted image, (e) addition of (d) and (b), (f) equalized image, (g) preprocessed image. **Source**: the author. 64
FIGURE 32 – Displays the result of each step of the algorithm 4, (a) preprocessed image, (b) blue channel, (c) green channel, (d) sum result of blue and green channels, (e) binarized image, (f) opening morphological operation, (g) closing morphological operation, (h) internal holes removed, (i) image with background removed. **Source**: the author. . 65
FIGURE 33 – Displays the result of each step of the algorithm 5, (a) background removed image, (b) hue channel, (c) binarized image using Otsu's method, (d) eroded image, (e) small objects removed, (f) multiplication result between the original image and the generated mask. **Source**: the author. 66
FIGURE 34 – Displays the result of each step of the Algorithm 6, (a) image composed with only leukocytes, (b) grayscale image, (c) binarized image, (d) watershed contours draws in (c), (e) border leukocytes flooded using the 127 value, (f) extracted border leukocytes, (g) subtraction of (c) minus (f), (h) morphological opening of (g) result, (i) image composed only of complete leukocytes. **Source**: the author. 68

FIGURE 35 – Examples of agglomerations in (a) and (b) and attached leukocytes in (c) and (d). **Source**: the author. 68
FIGURE 36 – Result of isolation method, (a) segmented leukocyte image, (b) image agglomeration, (c) attached image, (d) isolated leukocytes image. **Source**: the author. 71
FIGURE 37 – Nucleus segmentation pipeline results, (a) segmented leukocyte image, (b) Saturation channel, (c) b channel, (d) U channel, (e) result of sum of (c) and (d), (f) binarization of (e), (g) k-means applied on Saturation, (h) k-means image binarized, (i) AND logical operation applied on (f) and (h), (j) result of closing operation, (k) inverted image and small objects removed, (l) segmented image. **Source**: the author. 72
FIGURE 38 – Nucleus segmentation pipeline results, (a) Leukocytes defined as *attached*, (b) COM found using Algorithm 8 marked in red, (c) COM using complete object, marked in yellow, (d) COM found after combining and validating COM of (b) and (c), marked in green. **Source**: the author. 73
FIGURE 39 – Nucleus segmentation pipeline results, (a) COM marked in white by Algorithm 8 applied in the complete agglomeration image, (b) COM marked in red found using Algorithm 8 in the nucleus image, (c) COM using complete object, marked in green, (d) COM found after combining and validating COM of (a), (b) and (c), marked in blue. **Source**: the author. 74
FIGURE 40 – Sub-image segmentation method results, (a) sub-image extracted from ALL_IDB1 image, (b) background removed, (c) erythrocytes removed, (d) image (b) binarized with edges found by watershed drawn, (e) incomplete leukocytes removed by flooding extremities, (f) erosion applied on (e) to detach objects, (g) central leukocyte found, highlighted in gray, (h) all leukocytes removed except the central, (i) segmented image acquired by using (h) as a mask and multiply it by (a). **Source**: the author. 75
FIGURE 41 – Shows how many features of each method was used for the final feature vector. **Source**: the author. 77
FIGURE 42 – Displays segmentation results of the method described in section 4.5 on each subset and the complete ALL_IDB1 database. **Source**: the author. 80

FIGURE 43 – Exemplifies segmentation pipeline results in different lighting patterns, where the images on the second line (e, f, g and h) are the segmentation result of above images (a, b, c and d). **Source**: the author. 80
FIGURE 44 – Displays single leukocyte detection results from the method described in section 4.6 on each subset and the complete ALL_IDB1 database. **Source**: the author. 81
FIGURE 45 – Illustrates leukocyte individual COM detection results, blue contours represent *isolated*, green refers to *attached* and red contours identify *agglomerations*. (a) contains an error due to wrong attribution of what should be *attached* as *isolated* (black arrow). (b) image containing *neutrophil* which in this case has multiple COM, detected a loss of a single leukocyte (black arrow). (c) image with *agglomerations*, *attached* and *isolated* attributed correctly, and all leukocyte COMs computed correctly even with a *neutrophil* being present in this image. **Source**: the author. 82
FIGURE 46 – Central leukocyte segmentation results of method described in section 4.7 on ALL_IDB2 complete database and ALL_IDB1 sub-images extracted. **Source**: the author. 83
FIGURE 47 – Illustrates the test results done to visualize the impact of the feature vector size. **Source**: the author. 83
FIGURE 48 – Shows the accuracy of each classifier tested on development database. **Source**: the author. 84
FIGURE 49 – Results of the tests done to visualize the impact combine features extracted using different methods. **Source**: the author. 85
FIGURE 50 – Results of the *SVM* classifier in each database used. **Source**: the author. 85
FIGURE 51 – Segmentation result of the wrongly classified image. (a) original image Im107_0, (b) segmented image. **Source**: the author. 86

LISTA DE TABELAS

TABLE 1 – Haralick texture features 32
TABLE 2 – Resume of classification proposals 58
TABLE 3 – Parameter table for classifiers. 78
TABLE 4 – Accuracy of classification proposals compared to proposed method. **Source**: the author 87

LISTA DE ABREVIATURAS E SIGLAS

ALL	*Acute Lymphoblastic Leukemia*
LMA	*Leukocyte Median Area*
COM	*Center of Mass*
CNN	*Convolutional Neural Network*
ML	*Machine Learning*
ABRALE	*Associação Brasileira de Linfoma e Leucemia*
ANN	*Artificial Neural Network*
CADx	*computer aided diagnosis*
SVM	*Support Vector Machine*
LDA	*Linear Discriminant Analysis*
MLP	*Multilayer Perceptron*
KNN	*K Nearest Neighbor*
WBC	*White Blood Cells*

SUMÁRIO

1 INTRODUCTION 18
1.1 Motivation 19
1.2 Objectives 19
1.3 Challenges 19
1.4 Contributions 20
1.5 Document Structure 20
2 THEORETICAL BASIS 21
2.1 Medical Concepts 21
2.1.1 Blood Cells 21
2.1.2 Leukemia 22
2.1.3 Acute Lymphoblastic Leukemia 23
2.2 Computational Concepts 23
2.2.1 Digital Image Processing 24
2.2.1.1 Thresholding 24
2.2.1.2 Color Representation Models 24
2.2.1.3 RGB model 25
2.2.1.4 HSV Model 26
2.2.1.5 YUV Model 26
2.2.1.6 L*a*b Model 26
2.2.1.7 Mathematical Morphology 27
2.2.1.8 Structuring Element 27
2.2.1.9 Erosion and Dilation 28
2.2.1.10 Opening and Closing 28
2.2.1.11 Flood Filling 29
2.2.1.12 Center of Mass Estimation 29
2.2.1.13 Clustering Segmentation (K-means) 30
2.2.1.14 Watershed 30
2.2.2 Feature Extraction and Preprocessing 32
2.2.2.1 Gray-Level Co-Occurrence Matrix 32
2.2.2.2 Local Binary Patterns 33
2.2.2.3 Convolution Neural Network Feature Extraction 33
2.2.2.4 Z-Score Normalization 34
2.2.2.5 Recursive Feature Elimination 35
2.2.3 Machine Learning 35
2.2.3.1 Supervised Learning 35
2.2.3.2 Predictive Models 36

2.2.3.3 K-Nearest Neighbors 36
2.2.3.4 Linear Discriminant Analysis 37
2.2.3.5 Decision Tree 37
2.2.3.6 Perceptron 38
2.2.3.7 Multilayer Perceptron 38
2.2.3.8 Support Vector Machines 39
2.2.3.9 Accuracy 39
2.2.4 Software Libraries 40
3 STATE OF THE ART 42
3.1 Segmentation Proposals 42
3.1.1 Robust Segmentation and Measurement Techniques of White Cells in Blood Microscope Images 42
3.1.2 Segmentation of Microscopic Images for Counting Leukocytes 43
3.1.3 Segmentation of Leukocytes and Erythrocytes in Blood Smear Images . . . 43
3.1.4 Segmentation of Leukocytes in Blood Smear Images Using Color Processing Mechanism Inspired by the Visual System 44
3.1.5 A Leukocyte Nucleus Segmentation Scheme Based on Fingerprint Smoothing 45
3.1.6 An Adaptive Leukocyte Nucleus Segmentation Using Genetic Algorithm . . 45
3.1.7 Robust Leukocyte Segmentation in Blood Microscopic Images Based on Intuitionistic Fuzzy Divergence 46
3.1.8 Unsupervised Leukemia Cells Segmentation Based on Multi-space Color Channels 46
3.1.9 Segmentation of White Blood Cell from Acute Lymphoblastic Leukemia Using Dual-Threshold Method 47
3.2 Leukocyte Count Proposals 47
3.2.1 A Novel Auto-Segmentation Scheme for Colored Leukocyte Images 47
3.2.2 Leukocyte Nucleus Segmentation and Recognition in Color Blood-smear Images 48
3.2.3 An Automated Framework for Counting Lymphocytes From Microscopic Images 48
3.2.4 Automated Leukaemia Detection Using Microscopic Images 49
3.2.5 A Computer-Aided System for Differential Count from Peripheral Blood Cell Images 50
3.3 Classification Methodologies 51
3.3.1 Computer Based Acute Leukemia Classification 51
3.3.2 Morphological Classification of Blood Leukocytes by Microscope Images . . 51
3.3.3 Leukocyte Segmentation and Classification in Blood-smear Images 52
3.3.4 Automatic Morphological Analysis for Acute Leukemia Identification in Peripheral Blood Microscope Images 53
3.3.5 New Decision Support Tool for Acute Lymphoblastic Leukemia Classification 53
3.3.6 Identification and Classification of Acute Leukemia Using Neural Network . 54

3.3.7 Leukocyte Classification for Leukemia Detection Using Image Processing Techniques . . . 54
3.3.8 An Intelligent Decision Support System for Leukemia Diagnosis Using Microscopic Blood Images . . . 55
3.3.9 Naive Bayesian Classifier for Acute Lymphocytic Leukemia Detection . . . 56
3.3.10Leukocyte Classification in Microscopy Images for Acute Lymphoblastic Leukemia Identification . . . 56
3.3.11A Leukemia Diagnostic System Using Pre-Trained CNN's and a Classification Committee . . . 57
3.4 Discussion . . . 57
4 METHODOLOGY . . . 59
4.1 Image Databases . . . 59
4.1.1 ALL_IDB1 Database . . . 59
4.1.2 ALL_IDB2 Database . . . 59
4.2 Overview . . . 60
4.3 Development Base . . . 62
4.4 Redundant ALL_IDB2 Images Removal . . . 63
4.5 ALL_IDB1 Leukocyte Segmentation . . . 63
4.5.1 Background Removal . . . 63
4.5.2 Erythrocyte Removal . . . 65
4.5.3 Incomplete Leukocyte Removal . . . 66
4.6 Single Leukocyte Detection . . . 67
4.6.1 Discover Median Leukocyte Area . . . 69
4.6.2 Detach Leukocytes . . . 69
4.6.3 Isolate Different Classes . . . 70
4.6.4 Nucleus Segmentation . . . 70
4.6.5 Find Individual Leukocyte Centers of Mass on *Attached* . . . 71
4.6.6 Find Individual Leukocyte Center of Mass in Agglomerations . . . 72
4.6.7 Sub-image Extraction . . . 73
4.7 Sub-image and ALL_IDB2 Segmentation . . . 74
4.8 Feature Extraction and Selection . . . 75
4.8.1 Texture Features . . . 76
4.8.2 CNN Features . . . 76
4.8.3 Morphological Features . . . 76
4.8.4 Feature Vector . . . 76
4.9 Image Classification . . . 77
5 RESULTS AND DISCUSSION . . . 79
5.1 Segmentation Results . . . 79
5.1.1 ALL_IDB1 Leukocyte Segmentation . . . 79

5.1.2 Individual Leukocyte Detection 79
5.1.3 Sub-images and ALL_IDB2 Segmentation Results 81
5.2 Dimensionality Reduction 82
5.3 Isolated Features 84
5.4 ALL_IDB1 Classification 84
6 CONCLUSION 87
6.1 Future Works 88
6.1.0.1 Test the proposed method in other databases 88
6.1.0.2 Precise segmentation evaluation 88
6.1.0.3 Apply machine learning techniques in segmentation phase 88
6.1.0.4 Adapt the method to also count leukocytes 88
REFERENCES 90

1 INTRODUCTION

Leukemias are a group of cancers that manifest in the blood, more specifically in white blood cells (leukocytes). Leukemias that attack leukocytes are divided into two groups: myeloid and lymphoid, according to the leukocyte lineage that has mutated. There are also sub-groups of lymphoid and myeloid leukemia: chronic or acute. Leukemia is classified as chronic when there is an increase in mature cells, and when there is an increase in immature cells, it is classified as acute (MELO; SILVEIRA, 2013).

Acute lymphoblastic leukemia (ALL) is the most common form of cancer that manifests in children. However, 90% of the children who undergo treatment are cured. On the other hand, when manifested in adults, the cure rate decreases to 50%, according to the Brazilian Association of Lymphoma and Leukemia (ABRALE). Due to the fact that it is an acute leukemia, it drastically increases the number of young lymphocytes circulating in the peripheral blood, causing them to stop functioning properly and begin to reproduce in a disorderly way. In order for it to present a fast evolution in its clinical condition, it is essential that the diagnosis of ALL is done quickly, so that the patient can initiate the treatment, thus increasing his remission chances (PERINI, 2016).

The diagnosis of ALL is usually made based on the analysis of the patient's bone marrow obtained through a myelogram[1] sample. The sample collected on the myelogram is analyzed by pathologists and hematologists. After the application of a dye which highlights leukocytes from the other blood cells is performed the incidence, maturity and morphology of the lymphocytes present on the slice is verified (FARIAS; CASTRO, 2004).

Since patients with *ALL* suspicion must be diagnosed as soon as possible, it is important to develop tools that help doctors do so. In this context it would be interesting if a *computer-aided diagnosis (CADx)* were to be used, since this system helps to improve the accuracy of the diagnoses performed by a doctor, as well as provides another perspective of the case, and does not have its performance affected by stress, fatigue or distractions (AZEVEDO-MARQUES, 2001).

This work proposes a CADx system capable of segmenting and extracting a small compact representation, so as to classify blood smear images as positive or negative in the presence of *ALL*, using *digital image processing* and *machine learning* methods, serving as a tool to help pathologists diagnose patients with *ALL*.

[1] Exam in which a small portion of the patient's bone marrow is collected through a needle.

1.1 MOTIVATION

Based on the assumption, initially idealized by (SCOTTI, 2006), that manipulating images is easier than manipulating blood samples, systems that can extract and interpret information from blood smear images can be very helpful on the diagnosis support, providing a second opinion to doctors who analyze these images instead of the blood itself, which results on faster and more precise diagnoses. These quick diagnoses can have a considerable impact on the patients prognoses, so each tool capable of helping doctors has a potential of saving lives.

1.2 OBJECTIVES

This work aims to provide a pipeline capable of classifying blood smear images between carrier of *Acute Lymphoid Leukemia* and a *healthy* person, so the objectives to do so are:

- Elaborate a pipeline capable of segmenting leukocytes from a complete blood smear image, differentiating them from erythrocytes and platelets.
- Extract leukocytes present in images.
- Extract and select the best features to generate a compact representation of leukocytes.
- Classify individually all leukocytes from a blood smear images and then classify the complete image.

1.3 CHALLENGES

After research on the literature and some experiments, challenges that involved blood smear image classification were detected, and they are listed below:

- Blood smear images do not have a constant lighting, so the segmentation methods designed to do so must be able to deal with distinct lighting patterns.
- The size of blood cells is not fixed since there isn't a standard zoom level, and neither is it informed.
- Finding a compact representation that does not demand a huge computational power, as many images must be analyzed per day on a real case scenario.

1.4 CONTRIBUTIONS

The expected contributions by the proposed work are:

- Provide a method capable of segmenting blood smear images, without suffering considerable performance impacts, due to the presence of distinct lighting patterns.
- Analyze the impact of combining distinct kinds of features in order to form a feature vector.
- Present a suitable and compact representation of leukocytes.
- Indicate the most suitable classifier to this particular problem.
- Generate a comparison between an approach that extracts and classifies blood smear images based on global and local features.

1.5 DOCUMENT STRUCTURE

The next chapter, *Theoretical Basis*, is divided into two sections, the explain *medical* and *computational* concepts which are essential for a complete understanding of the proposed work.

The *State of the Art* presents works related to this same context, being divided into three main topics, according to the objective of the proposals: leukocyte segmentation, blood cell count and *ALL* detection.

Chapter four describes the *Methodology* proposal, describing the pipeline used to segment, extract features and classify blood smear images, explaining and justifying each step of the pipeline proposed.

Chapter five focuses on presenting the results of each step, starting with segmentation, then feature selection and combination, to finally end with classification results, which is the main objective of the work.

Finally, the *Conclusion* chapter ends the proposal by presenting an explanation of what can be extracted from the obtained results, a comparison of the works described in *State of the Art* focused on the classification and some proposals for possible future works.

2 THEORETICAL BASIS

This chapter presents medical and computational concepts, which are necessary to understand the proposed work, by starting to introduce medical concepts about blood cells and leukemia in Section 2.1, then presenting computational concepts in Section 2.2. Computational concepts were divided into four sub-sections: *Digital Image Processing* 2.2.1, *Feature Extraction Methods* 2.2.2, *Machine Learning* 2.2.3 and *Software Libraries* 2.2.4.

2.1 MEDICAL CONCEPTS

Since this work intents to develop a medical solution using computer science techniques, understanding about ALL is crucial for the development of an optimal solution. Since ALL is an illness that manifests itself in the blood, it is essential to understand which kind of cell is affected, which differentiates a healthy patient from a sick one.

2.1.1 Blood Cells

Blood cells are separated into three groups: *erythrocytes*, *platelets* and *thrombocytes*. Each one has a different specialization, ranging from promoting coagulation to defending the body from infections. Even with a different specialization, morphology and maturation cycle, all blood cells have a common origin in stem cells in red bone marrow. It is during the maturation cycle, described in Figure 1, that a stem cell specializes itself until it becomes an *erythrocyte*, a platelet or a leukocyte.

Red blood cells or erythrocytes are anucleated cells in the shape of a biconcave disc, and their responsibilities go from delivering oxygen from the lungs to the entire body, and carrying carbon dioxide back to the lungs, so it can be expelled from the body (HUTTER; STÖHR, 1982).

Different from erythrocytes and leukocytes platelets or thrombocytes, aren't cells but cell fragments originated from megakaryocytes cytoplasm, and each megakaryocyte generates around 4000 and 5000 thrombocytes (DJALDETTI et al., 1979). Thrombocytes participate in the process of blood clotting, retraction and the removal of blood clots.

White blood cells or leukocytes compose the immune system defending the body against virus, bacteria, fungi and parasites. There are five classes of leukocyte which are: *neutrophil*, *eosinophil*, *basophil*, *monocyte* and *lymphocyte*, each one with a different nucleus morphology and speciality, illustrated in Figure 2. Leukocytes are divided in two main groups based on the existence of granules in their cytoplasm which are granulocytes

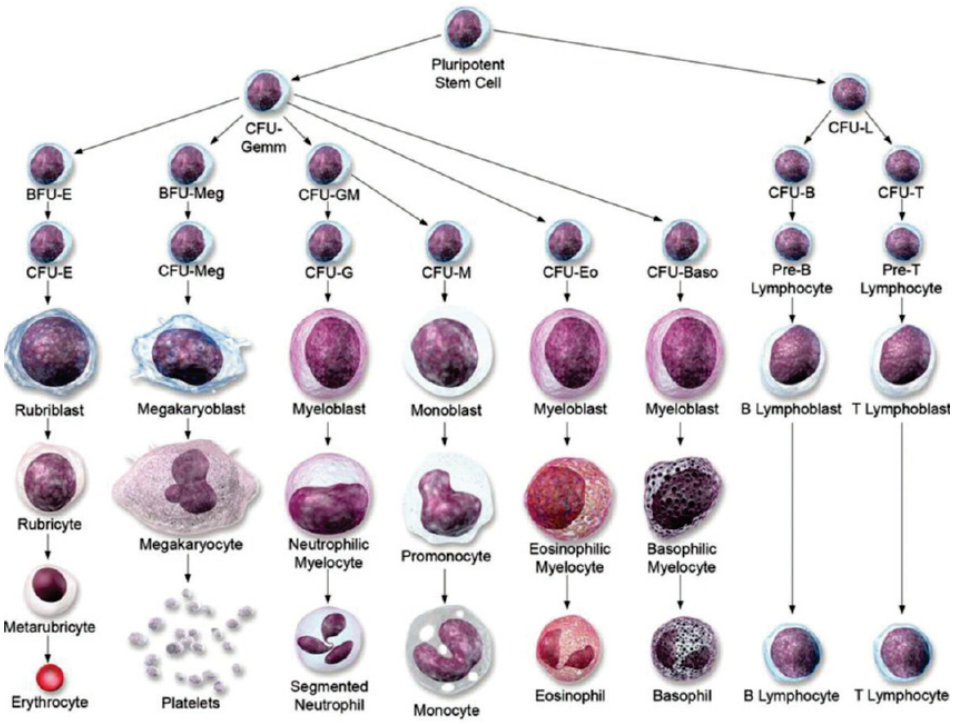


FIGURE 1 – Erythrocytes in a blood smear, where a nucleus absence and the disco shape can be noticed. **Source**: (BARRETO et al., 2014)

and agranulocytes, where lymphocytes and monocytes are agranulocytes and bashophiles, eosinophiles and neutrophiles are granulocytes.

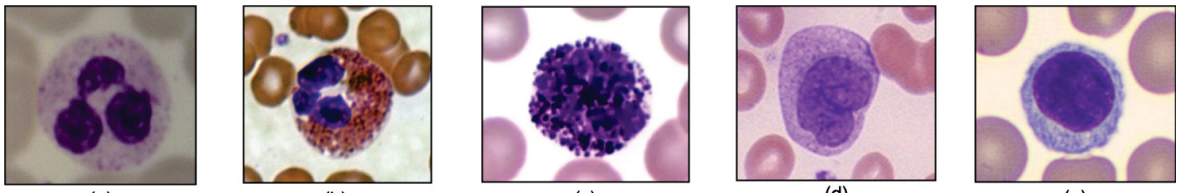


FIGURE 2 – Leukocyte classes, (a) neutrophil, (b) eosinophil, (c) basophil, (d) monocyte, (e) leukocyte. **Source**: the author

2.1.2 Leukemia

Leukemias are the most common types of cancer that manifest in blood cells, changing leukocytes, stopping them from accomplishing their function and reproducing themselves in a disorderly fashion, compromising all blood functions, but mostly the immunologic system of the carrier.

There are two kinds of leukemia that can develop in a patient, where the *acute* one generates a fast growth of immature cancer cells, and the *chronic* one is characterized by the elevation of the number of mature cancer cells (HAMERSCHIAK, 2016b).

The most known kinds of leukemia are *myeloid* and *lymphoblastic*, and what distinguishes them is the leukocyte lineage that is affected. *Myeloid* affects all granulocyte

groups (eosinophil, basophil and monocyte), while *lymphoblastic* affects only the lymphocyte lineage. All blood functions of the patient are compromised, exposing him/her to all kinds of infection because the immunologic system does not work. Because the erythrocytes are also affected, the patient becomes anemic, and acquires other symptoms such as bone pain, discomfort on the left side of the abdomen due to swelling of the spleen and nausea (WOELFEL, 2017).

2.1.3 Acute Lymphoblastic Leukemia

Acute Lymphoblastic Leukemia (ALL) is the most common kind of cancer that manifests in children, and for being acute this illness evolves vary fast, so it is very important that the diagnoses is made as soon as possible, thus improving the chance of remission.

Usually the specialist suspects that his patient has leukemia when he/she is sick all the time and has anemia, since those are classic symptoms of leukemia, the first test to be done is a complete hemogram, because it shows the count of each blood cell, so if the patient has leukemia, the results will be very different from the reference value. This test also shows the count of each leukocyte lineage pointing to the leukocyte lineage being affected (HAMERSCHIAK, 2016a). If the result of the complete hemogram confirms that the patient has alterations, the specialist usually does a lumbar puncture to see on the microscope how the leukocyte maturation process is evolving. Based on this test, it can be confirmed if the patient has leukemia or nor, and if it is lymphocytic or monocytic.

The standard treatment for ALL is chemotherapy, with a response success rate of close to 90% in children(HAMERSCHIAK, 2016a). When chemotherapy is not enough to control the disease or when the patient's condition is already very advanced, a bone marrow transplant is indicated. A bone marrow transplant is suggested only in specific cases due to the risk of graft-versus-host disease that can happen because of donor and patient immunological incompatibility (SILVA et al., 2005).

2.2 COMPUTATIONAL CONCEPTS

In each step of this project computational concepts were used from the segmentation phase to the classification phase, and the complete understanding of these concepts are essential to use these techniques in the best way. The computational concepts used were divided into two groups, according to their research area: *Digital Image Processing* or *Machine Learning*.

2.2.1 Digital Image Processing

As one image can be defined as a dimensional function *f (x, y)*, where x and y are spatial coordinates and f is the intensity of the coordinate, *Digital Image Processing* is any process that changes the intensity of one or more pixels (*coordinate(x, y)*) for a specific objective (GONZALEZ; WOODS, 2010).

2.2.1.1 Thresholding

An ordinary operation used in *digital image processing* is to binaryze a grayscale image for a specific objective. This operation is done using a method known as *sample threshold*. This operation consists in defining a value T and verifying all pixels on the image, by applying Equation 2.1 in order to define the new value of each pixel, thus more than one T value can be defined. Threshold operations are widely used in segmentation processing (ROSEBROCK, 2016).

$$threshold(pixel\_value) = \begin{cases} 255, & \text{if } pixel\_value > T \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

A problem can be found when observing how basic threshold works, that is, which value must be attributed to the T segment, only with the image portion wanted, and a more important issue is that a fixed value attributed to T probably won't work well on all images of the database, as the *basic threshold* method suffers a major impact with lighting changes (FILHO; NETO, 1999).

A method used to find the T value automatically was developed by (OTSU, 1979), assuming that there are two peaks on the image histogram, where each peak is attributed as one class and the vale which divides these peaks contains the optimal value for T. In order for Otsu's method to work well, the image histogram must have the two valleys well defined, and in case that this isn't true, a preprocessing on the image to adapt it to this premise is acceptable. Otsu's method is less sensitive to illumination changes to the *sample threshold* since the T value is defined based on statistical characteristics of the image itself, as illustrated on Figure 3.

2.2.1.2 Color Representation Models

A color representation model also called color space or color system is a specification of a coordinate system, where each point is a distinct color (GONZALEZ; WOODS, 2010). There are color representation models idealized to work directly with the hardware as printers (CMYK) or screens (RGB), and models that were designed to represent the colors as humans see them (HSV) or in representations that facilitate the processing by computers (L*a*b).

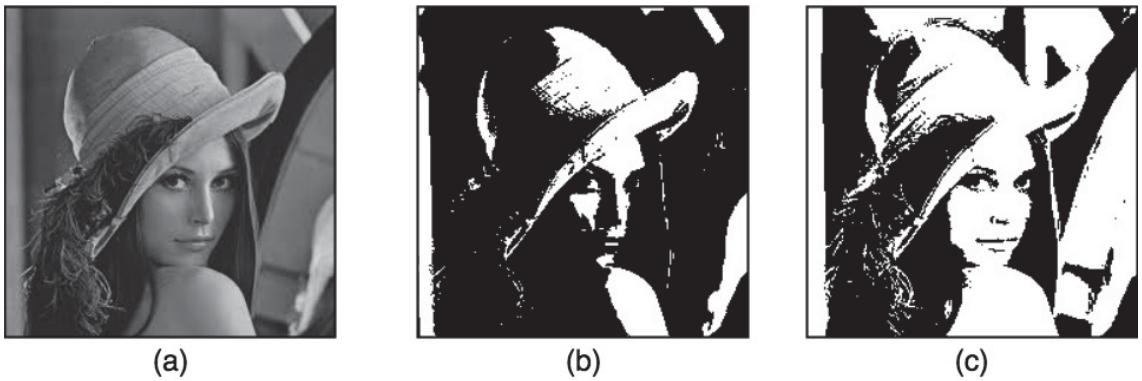


FIGURE 3 – (a) original grayscale image, (b) result of the sample threshold application using T as 127, (c) result of Otsu's method. It is observed that image (c) preserved many more details of the original image than image (b). **Source**: the author

2.2.1.3 RGB model

The Red Green Blue (RGB) model is an additive model that bases itself on Cartesian coordinates where each point represents a color made by the color combination of red, green and blue, making it possible to interpret this model as a cube, illustrated on Figure 4. As RGB is an additive model, any RGB image is in fact made of three grayscale sub-images, one for each primary channel, that when combined, results on the colored image (GONZALEZ; WOODS, 2010). The value of each pixel varies between zero (complete absence), resulting in the color black, and the biggest intensity (usually 255), resulting in the color white.

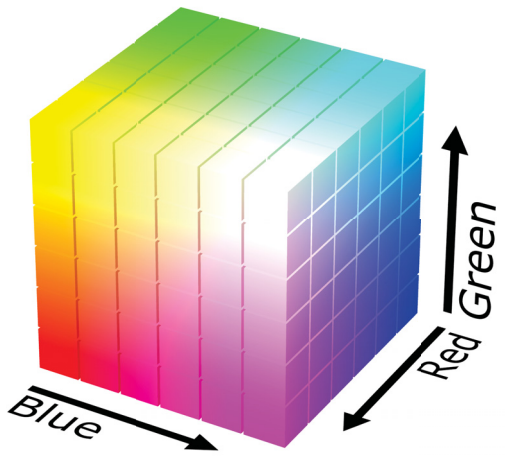


FIGURE 4 – The RGB model interpreted as a three-dimensional cube. **Source**: <https://upload.wikimedia.org/wikipedia/commons/a/af/RGB_color_solid_cube.png>

2.2.1.4 HSV Model

The Hue Saturation Value (HSV) model represents color like humans perceive them. Hue represents the purity level, used to differ orange from red where red has a higher hue level than orange. Saturation refers to dilution level of a pure color on white light, and the value channel represents brightness. The HSV model is largely used on digital image processing systems for isolate color intensity, which shows itself to be a good descriptor in the differentiation of distinct objects (ALVES, 2010). Geometrically, this model is represented by a hexagonal pyramid (Figure 5).

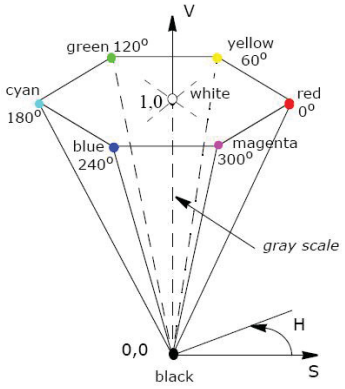


FIGURE 5 – Geometric representation of the HSV model. **Source**: <https://software.intel.com/en-us/node/503873>

2.2.1.5 YUV Model

The color model developed to be applied in the transmission of television signals, this model was developed to cover the need for a model that could display colorful images and black and white in an efficient way, since there were televisions that still did not have support for colored images and transmitting two different signals would be very costly. Thereby the luminance, which is the luminosity perception and brightness represented on the Y-channel were separated from the color information, allowing the TV, which did not support color images, to display the luminance channel (OLIVEIRA, 2007).

2.2.1.6 L*a*b Model

The L*a*b color representation model was based on the opposite color theory, where two colors can't be green and red at the same time, or yellow and blue at the same time. In this model, colors are represented on channels a and b, while the L channel measures the luminosity of the image. The a channel defines if the color is closer to green or yellow, and the b channel defines if the color is closer to blue or yellow (INTEL, 2010).

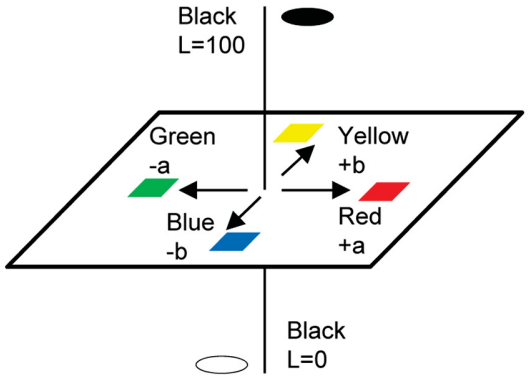


FIGURE 6 – L*a*b color model representation, illustrate the opposite color theory. **Source**: (LIU et al., 2014)

2.2.1.7 Mathematical Morphology

Mathematical morphology is the digital image processing research field devoted to work with the shape of the objects that compose an image, its theoretical basis comes from mathematical set theory. Mathematical morphology operations are usually applied on binary images with a distinct result like: enhancement, filtering, segmenting, and edge detection (FILHO; NETO, 1999).

2.2.1.8 Structuring Element

A basic concept used in morphological operations is the structuring element or kernel, which is defined as a set or a rectangular mathematical arrangement, used as base to all morphological operations. The most important characteristic of a structuring element is its reference point and the value related to the other elements.

The reference point is indicated with an x in the examples illustrated in Figure 7, it's used to position the structuring element during the convolution process. The other values of the kernel defines the pattern or shape that will be applied on the image with the objective to change the shape of the objects. (FILHO; NETO, 1999)

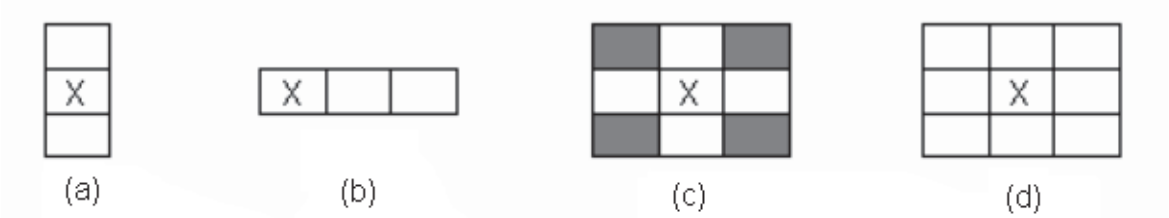


FIGURE 7 – Structuring element examples, with distinct shapes. **Source:** <https://sofaltatestar.wordpress.com/2010/01/06/morfologia-matematica-para-imagens-em-tons-de-cinza/>

2.2.1.9 Erosion and Dilation

The most basic mathematical morphology operations are erosion and dilation, both work by convolving a structuring element B on an image A, where during the convolution, a kernel reference point neighborhood to define its value on the resulting image is verified.

In an erosion morphological operation, the set relative to the structuring element must be completely contained in the region of the pixel with the same coordinates so that the kernel reference point is activated on the resulting image. The application of a erosion makes the elements smaller, being useful when the objective is remove objects smaller than the kernel, increase the holes and separate the connected components.

In a morphological dilation operation it is enough that only one activated element, belonging to the structuring element, overlays a value also activated on the image, for that pixel also to be activated on the resulting image. This operation results in the elimination or reduction of holes and particles in the connection of nearby regions. Figure 8 shows the results of the application of morphological erosion and dilation operations (SOLOMON; BRECKON, 2013).

FIGURE 8 – Illustrates the result of an erosion and dilation morphological operation application. (a) original image, (b) erosion result, (c) dilatation result. **Source**: <https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html>

2.2.1.10 Opening and Closing

In some cases there is the necessity to preserve the original object shapes, as much as possible. Thus using only an erosion or a dilation operation, it is inevitable that the resulting image will have considerable morphological changes in its object shapes, these changes are directly related to the structuring element shape and size used, the bigger it is, the bigger the changes.

When combining the two basic operations (erosion and dilation), the resulting image information loss is reduced, and depending on the result wanted an erosion is applied, rather than a dilation (opening) or a dilation, rather than an erosion (closing). An

opening operation is used to smooth outlines and remove small objects, while a closing is used to connect nearby components and the closing of small holes (GONZALEZ; WOODS, 2010). Figure 9 shows the result of an opening and closing operation making it possible to notice the difference between these operations using a kernel 5x5.

FIGURE 9 – Displays the results of opening and closing operations on a noised image, where (a) is the result of an opening used to remove small objects, and (b) is the result of a closing operation used to close all small holes on the object. **Source**: <https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html>

2.2.1.11 Flood Filling

The method known as region growing or flood filling is utilized on image segmentation to fill holes and contours, and it is usually applied on binary images, but it can also be used on grayscale images with some changes. Through an initial coordinate, also know as seed, the flood filling method consists of, verifying its neighborhood pixels and changing the values of those that comply with a predefined condition, which usually is a threshold function changing the value of pixels with its value being bigger or lower than a reference value, that meets a predefined condition has its value changed (GONZALEZ; WOODS, 2010). The delicate point that involves this method is that the seed point must be informed, so this parameter must be known before the flood filling application.

2.2.1.12 Center of Mass Estimation

Image moments are a group of statistical measures used to describe an image, and by using some of the statistical moments it is possible to estimate the center of mass or gravity center of an object. Statistical moments of zero order and statistical moments of first order are used in the mass calculation center, consisting in dividing the first order moments m_{10} and m_{01} by the moment of zero order m_{00} to find the coordinates of the mass center, described in Equation 2.2, where a statistical moment is computed using Equation 2.3 (BARELLI, 2018).

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad and \quad \bar{y} = \frac{m_{01}}{m_{00}} \tag{2.2}$$

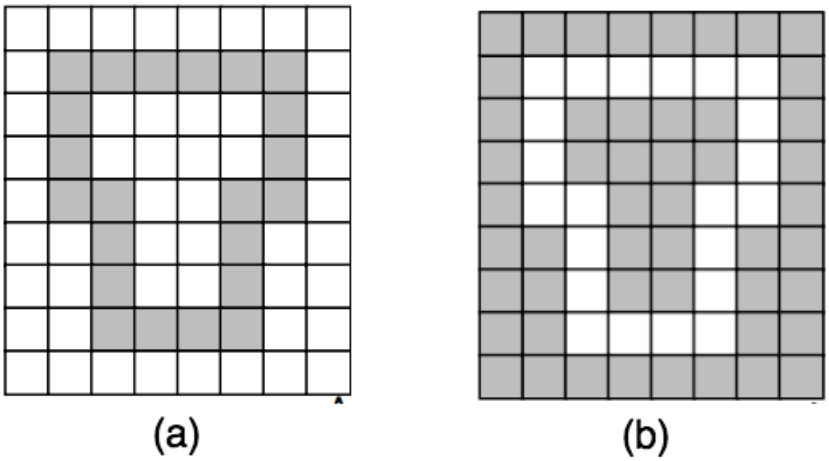


FIGURE 10 – (a) Binary image composed only of contours; (b) Result of region growth, applied in image (a), using at least one external point and another internal one for the contour as seed points. The contours have their value inverted for illustrative purposes. **Source**: (FILHO; NETO, 1999)

$$m_{p,q} = \sum_x \sum_y x^p y^q I(x, y) \tag{2.3}$$

2.2.1.13 Clustering Segmentation (K-means)

Clustering segmentation is a method that groups pixels with a similar or little difference and attributes a new value to each pixel group, resulting in a decrease of possible pixel values, where the number of clusters are defined by the parameter K. A more detailed K-means procedure is described in Algorithm 1.

Algorithm 1 K-means algorithm:

1: Define K value
2: Place all K centroids randomly
3: Group neighborhood pixels with close values to its nearest centroid
4: Recalculate centroid coordinates, where a centroid must be at the center of its cluster, after attributing every pixel to a centroid
5: Repeat steps 3 and 4 until step 4 doesn't result in any changes

The ideal value of K varies according to each case, and in order to do that, a prior knowledge of what exists on the image is usually needed. So the sensitive point of this method is which value will be attributed to K in order to accomplish the objective without losing relevant information, as illustrated in Figure 11 (GATH; GEVA, 1989).

2.2.1.14 Watershed

Watershed (BEUCHER; MEYER, 1993) is a classic technique used on contour detection. It consists in interpreting a grayscale image as a topographic representation,

FIGURE 11 – Shows result variation of the K-means algorithm application with distinct values attributed to K. **Source**: <http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_ml/py_kmeans/py_kmeans_opencv/py_kmeans_opencv.html>

where pixels with higher intensity are interpreted as higher places in the relief than pixels with lower intensity. The following step is to "flood"lower relief places evidencing high relief locations resulting on contour detection, as illustrated in Figure 12.

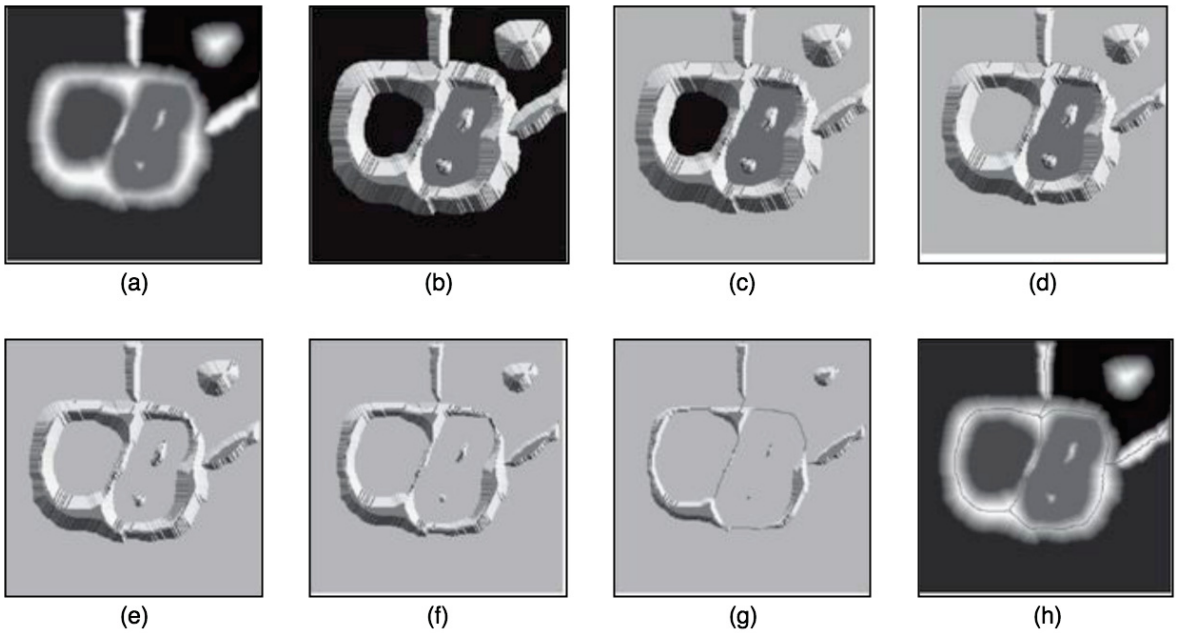


FIGURE 12 – Displays the phases of watershed algorithm applied on image (a); (b) is the topographic representation; images (c) to (g) contain the results on several floods; (h) contains the final watershed contours. **Source**: (GONZALEZ; WOODS, 2010)

2.2.2 Feature Extraction and Preprocessing

When working with a classification problem, a way is normally used to find a numerically representation that describes the objects that will be classified (SILVEIRA; BULLOCK, 2017). The ideal scenario is to find the smaller set of features that can distinguish the classes involved. Once more features are used more, computational processing power is needed and more data must be used in classifier training.

There are several methods used to extract features from images. In this section, texture descriptors (*GLCM* (HARALICK et al., 1973) and *LBP* (OJALA et al., 1996)) and Convolutional Neural Network (CNN) are described. Feature extraction like this one was the method used to build the feature vector in this work.

2.2.2.1 Gray-Level Co-Occurrence Matrix

Gray-Level Co-occurrence Matrix consists of a symmetrical matrix with the occurrences of grayscale pair values. It is usually computed using a single direction as reference to match the pairs, and it is possible to use two or more directions simultaneously, but in this case the final *GLCM* is computed from the average of each matrix direction.

The *GLCM* position contains the occurrence number of a determined pattern. The pattern location is related to its position, e.g.: in position *(i, j)* it is the number of times that a pixel with value i is adjacent to a pixel with value j. Thirteen statistical measures, which have a great description of the image which are extracted by using *GLCM* (HARALICK et al., 1973).

TABLE 1 – Haralick texture features

Feature	Description
f1	Angular Second Moment
f2	Contrast
f3	Correlation
f4	Sum of Squares: Variance
f5	Inverse Difference Moment
f6	Sum Average
f7	Sum variance
f8	Sum Entropy
f9	Entropy
f10	Difference Variance
f11	Difference Entropy
f12	Information Measure of Co-relation 1
f13	Information Measure of Co-relation 2

2.2.2.2 Local Binary Patterns

Local Binary Patterns *(LBP)* (OJALA et al., 1996) is a feature extraction method that approaches images in a structural and statistical way, and it is based on the hypotheses that the binary patterns of a pixel circular neighborhood is a fundamental texture characterization. The feature vector extracted from the LBP is nothing more than the histogram of the occurrence of each of the 256 possible LBP codes.

The *LBP* code calculation consists of convolving a mask with radius R and a number of pixels P in a grayscale image, where each iteration consists in positioning the mask then applying a threshold in the region, using the central pixel value as a threshold, which changes each of the region values to 0 or 1. Then the threshold resulting image is multiplied by another mask composed only of power of 2 values. Finally, all values of the multiplication resulting matrix, which will give the *local binary pattern code* of this pixel are added, as illustrated in Figure 13.

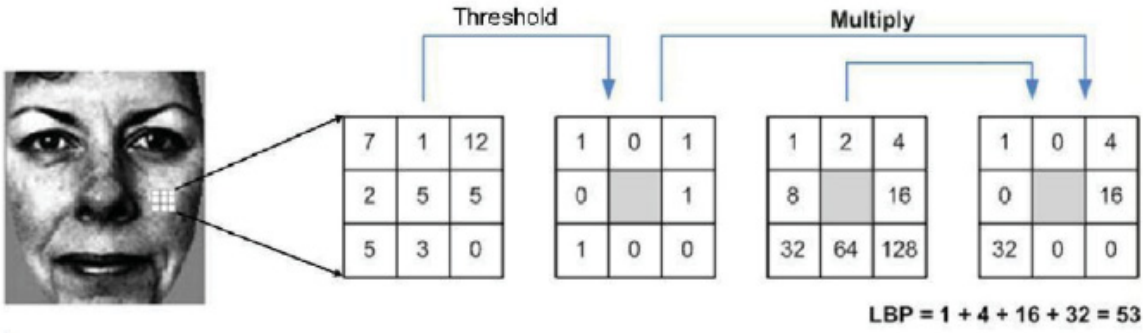


FIGURE 13 – Local Binary Pattern calculation example. **Source**: (GORODNICHY et al., 2014)

2.2.2.3 Convolution Neural Network Feature Extraction

Convolutional Neural Network is a neural network based algorithm that is becoming the state of the art in many image processing problems. It consists of a neural network that uses an image on the input layer and has a convolution and pooling operations between its layers, thus reducing the image size and generating a higher level representation after each layer. The final layer, also known as fully connected layer, is a multilayer perceptron neural network used to classify the image, as illustrated in Figure 14.

The first architecture presented was the LeNet (LECUN et al., 1998), used in a hand written digit recognition and presents very significant results. Microsoft researchers developed a technique known as *residual learning* (HE et al., 2015), where the idea is there aren't huge changes between representation maps of close layers, so that after convolving and pooling operations, the resulting representation map is added to the previous one,

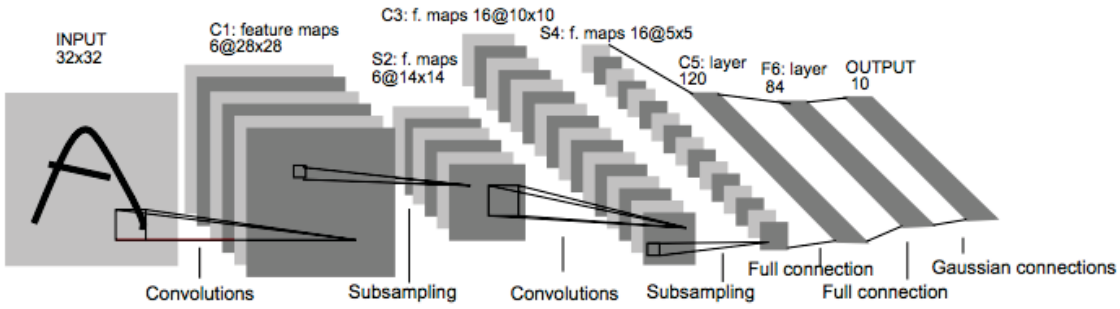


FIGURE 14 – Architecture of the CNN LeNet-5. **Source**: (LECUN et al., 1998)

enhancing its characteristics after each layer. Figure 15 shows the architecture of a *residual block*.

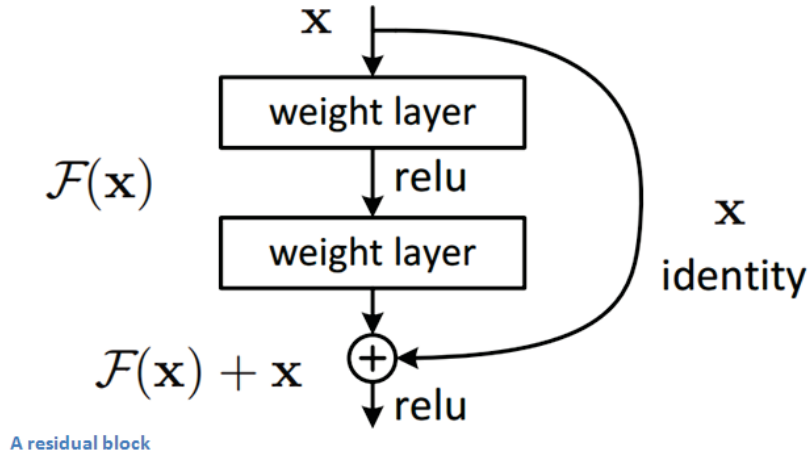


FIGURE 15 – Residual block architecture. **Source**: (DESHPANDE, 2016)

The problem faced when working with CNNs is the necessity of a large volume of images needed for training. one way to work around it when there aren't many images available, is generating artificial data by enlarging the dataset (WONG et al., 2016). Another way to use CNN, when working with a small dataset is using a pretrained CNN as a feature extractor, where instead of converging and using the multilayer perceptron on the final classification, by stopping the process earlier, using the extracted representation (RAZAVIAN et al., 2014).

2.2.2.4 Z-Score Normalization

As the distinct feature extraction methods can generate feature values in different scales, feature normalization is an almost mandatory step in the *Machine Learning* systems. Some predictive models can assign a wrong bias to features with bigger values, giving it more weight than it should. Also, it is easier to deal with outliers on normalized data (FACELI et al., 2011). The *Z-score* normalization method uses the average (μ) and variance

(σ) to define the new value of a sample (υ), as detailed in Equation 2.4. *Z-score*.

$$\upsilon New = \frac{\upsilon - \mu}{\sigma} \tag{2.4}$$

2.2.2.5 Recursive Feature Elimination

As the ideal scenario in every ML algorithm is to differentiate objects using the least possible features, the feature selection or dimensionality reduction is a very important preprocessing step. *Recursive Feature Elimination (RFE)* (GUYON et al., 2002) is an algorithm used in feature selection. It works in a loop, where at each iteration a linear regression model is used to estimate the less important feature, and then remove it. The loop runs until a parameter n, which defines the number of features that will be reached, detailed in Algorithm 2. There are two concerns when *RFE* is used. The first one is which value will be defined to n, usually found running several times until the classification phase stabilizes, and the second one is that it needs greater computational resources needed increase according to the initial number of features that will be selected.

Algorithm 2 Recursive feature elimination algorithm:

```
1: Define final number of features n
2: while Feature vector size > n do
3:     Train a linear regression model (SVM for example)
4:     Verify the importance of each feature and select the less important index
5:     Remove the less important feature
6: end while
```

2.2.3 Machine Learning

Machine Learning (ML) is a computer science research area that studies the capacity to improve the performance of some tasks by using previous experience (MITCHELL, 1997). Usually the ML systems are used in prediction tasks, where a predictive model is trained using a dataset to predict if a new sample belongs to class A or B. A predictive model training phase consists in analyzing the available dataset and finding a *decision function* that discriminates objects that belong to distinct classes (SILVEIRA; BULLOCK, 2017).

2.2.3.1 Supervised Learning

The method known as supervised learning consists of working with data previously labeled, so when a feature vector is passed on to a classifier during its training phase, the label relative to its class is also informed. When problematic classes are not known or the data isn't labeled, an approach known as unsupervised learning can be used, which tries to find clusters that share similarities and define each cluster as a class (HTIKE; KHALIFA, 2010).

2.2.3.2 Predictive Models

A predictive model is an algorithm that, based on an amount of data can find a function which differentiate samples of different classes (FACELI et al., 2011), depending on the algorithm used to find a decision function the classifier can be defined as linear or nonlinear, where linear classifiers generate a linear decision boundary that divides classes, and nonlinear classifiers generate a non-linear decision boundary, illustrated in Figure 16.

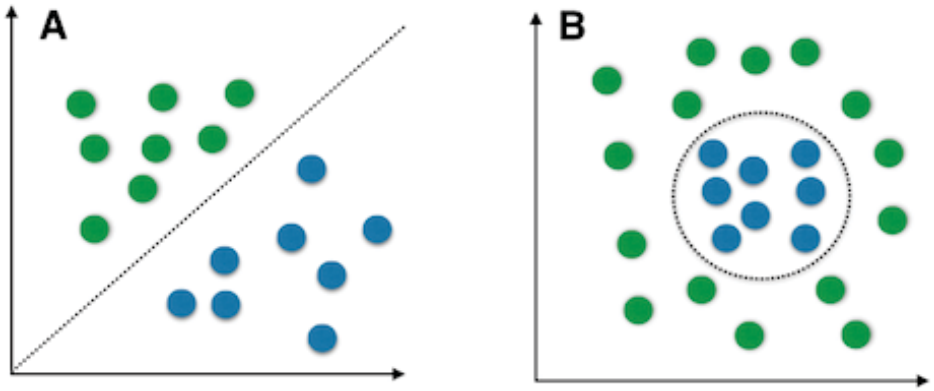


FIGURE 16 – It displays a boundary that splits dots of different classes, according to its color in two different scenarios, where there is a linear boundary on A and a nonlinear boundary on B. **Source**: <https://i.stack.imgur.com/OrcTJ.png>

2.2.3.3 K-Nearest Neighbors

The algorithm *K-nearest neighbors (KNN)* is a predictive model based on the hypothesis that similar data must be concentrated in the same zone, where distinct data are distant of each other. *KNN* is considered a lazy predictive model, due to the fact that it does not learn a compact representation to distinguish objects from different classes. Instead, it calculates the distance (Euclidean) between all objects on the training dataset and the new sample. Because of that, *KNN* can become inefficient when applied to large amounts of data (FACELI et al., 2011).

Parameter K defines the number of examples belonging to the training dataset that will be used to define which class the new sample belongs to. Usually odd values are used so that there aren't any draws. That is important, since the class with bigger apparitions is inferred to the new object.

Although *KNN* is a very simple algorithm, it can perform very well on problems with complex boundaries, in this case, beating linear classifiers much more complex which indeed learn a representation model (AHA et al., 1991).

2.2.3.4 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) (FISHER, 1938) is a generalization of *Fisher's linear discriminant* that uses statistical methods to change the data projection and find the best space that divides data of distinct classes. It became popular after the publication of the article *Eigenfaces vs Fisherfaces: Recognition Using Class Specific Linear Projection* (BELHUMEUR et al., 1997), once the results presented by *LDA* on the face detection problem where very good.

LDA seeks a linear transformation by minimizing the intraclass distance and maximizing the distance between distinct classes. When applied in linear separable problems, *LDA* rotates data, seeking that when classes start to be projected on a plan they can be separated in an easier way, as observed in Figure 17.

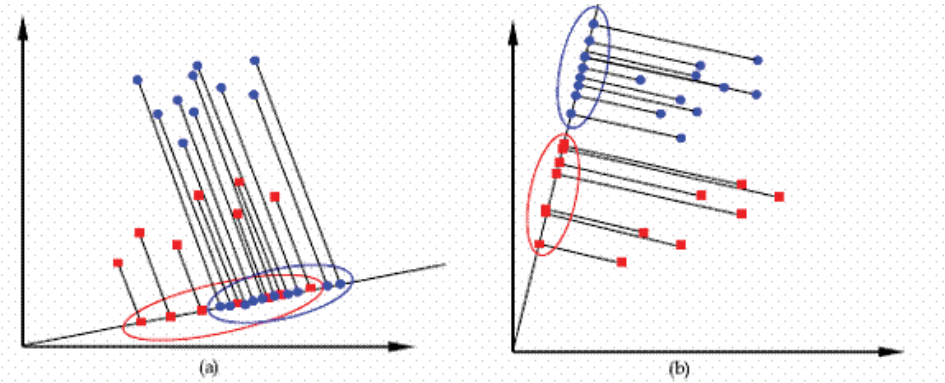


FIGURE 17 – Displays the result of the *LDA* data transformation by rotating it. **Source**: <http://www.tutorial.freehost7.com/human_face_recognition/linear_discriminant_analysis.htm>

2.2.3.5 Decision Tree

Decision tree is a machine learning algorithm based on a search widely used on diagnoses and risk analysis systems. It uses the strategy of *dividing and conquering*, in order to split a complex problem into less complex small problems, thus finding the solution more easily (FACELI et al., 2011). Some particularities of this method, as they can deal with numerical and symbolic data, don't need data normalization and is less sensitive to noisy data in the training phase, which makes it very powerful in some cases.

Two fundamental concepts involved in decision tree algorithm is *leaf* and *branches*, where a *branch* is a decision function that based on an input value decides which branch or leaf is the next step, and *leaf* is the final value found by a number of previous decisions (LUGER, 2013).

2.2.3.6 Perceptron

Proposed by Rosenblatt in 1958, *Perceptron Neural Network* (ROSENBLATT, 1958) was the first Artificial Neural Network (ANN) architecture, illustrated in Figure 18. Although this architecture is composed by only one artificial neuron, it presents relevant results for many classification problems. Rosenblatt proved the theorem of artificial neural network convergence, mathematically concluding that *if it is possible to classify a set of inputs linearly, then a Perceptron can do it* (HAYKIN, 2001).

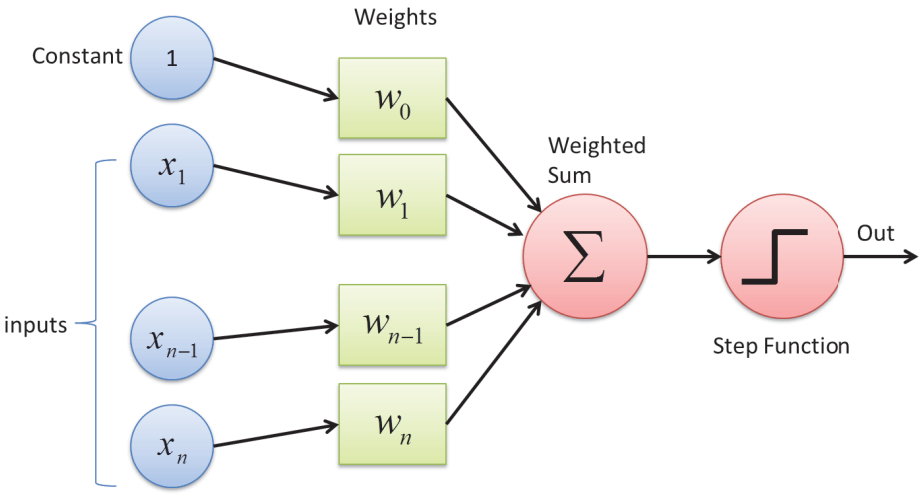


FIGURE 18 – Displays Perceptron architecture. **Source**: <http://abhay.harpale.net/blog/machine-learning/a-hands-on-tutorial-on-the-perceptron-learning-algorithm/>

Perceptron training phase consists of iterating each of the training samples, classifying them and adjusting the weights when needed, until the entire training base or at least a determined limit of the training samples classified correctly are reached. The algorithm used to readjust the Perceptron input layer weights is known as *backpropagation* (HECHT-NIELSEN, 1989). The perceptron's limitation comes when facing nonlinear problems, since a Perceptron ANN can find only linear separations.

2.2.3.7 Multilayer Perceptron

As the majority of the classification problems tends to be non linearly separable, using linear classifiers as a Perceptron won't solve the majority of the problems, as this classifier can find only linear boundaries. A classifier capable of dealing with non linearly separable problems is the *Multilayer Perceptron (MLP)*. It consists of adding one or more layers between the input layer and the output neuron (HAYKIN, 2001), illustrated in Figure 19.

The hidden layer is interpreted as an extra feature extractor phase, and it can have several neurons to find several boundaries, theoretically being possible to solve any complex classification problem. However, the more layers and neurons the net has, the

more data must be available to train the net, and this happens because there are more weights to be balanced (LUDWIG; MONTGOMERY, 2001).

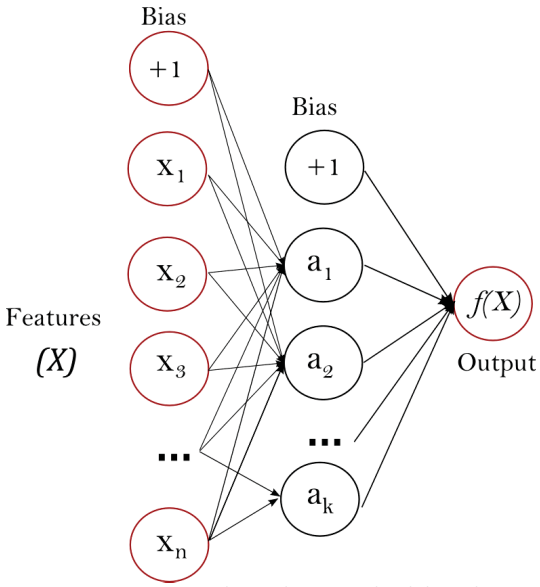


FIGURE 19 – *Multilayer Perceptron* with only one hidden layer. **Source**: <http://scikit-learn.org/stable/modules/neural_networks_supervised.html>

2.2.3.8 Support Vector Machines

Support Vector Machines (SVM) is a predictive model that bases itself on the statistical learning theory, proposed by Vapnik in 1995 (CORTES; VAPNIK, 1995). This method presents very consistent results, even beating MLP in several cases. Vapnik's discovery is the kernel function used to increase feature space, since to each an every problem there is a space that contains a linear division of the dataset, even if it is in a space with more dimensions (FACELI et al., 2011).

Different from Perceptron, which always finds a decision boundary if the dataset is linearly separable, this boundary is not necessarily the best one the biggest generalization possible. The SVM boundary is the best one possible. The boundary defined by SVM has a high generalization power because it presents the same distance to the margin of the two distinct classes, illustrated in Figure 20.

2.2.3.9 Accuracy

Accuracy is a measure used to evaluate a classifier, that tells how many of the samples were classified correctly, by dividing the sum of the positive samples (TP) and the negative samples classified correctly (TF) by the total of samples used in the test ($TOTAL$) as detailed in Equation 2.5. The problem of using only accuracy to define is a classifier is performing well if that this measure doesn't give a clue as to which class the

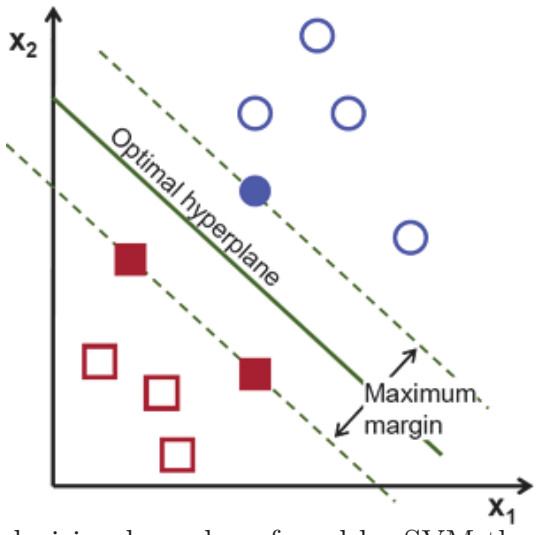


FIGURE 20 – Optimal decision boundary found by SVM through a hyperplane based on the margins of each class. **Source**: <http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html>

classifier is more sensitive. In some problems is very important to know this (FACELI et al., 2011).

$$Accuracy = \frac{TP + TF}{TOTAL} \tag{2.5}$$

2.2.4 Software Libraries

As this work uses methods of *digital image processing* and *machine learning*, libraries that contain the implementation of these methods were searched. the following five libraries were widely used in this work: *OpenCV*, *mahotas*, *scikit-learn*, *NumPy* and *keras*.

Open Source Computer Vision Library OpenCV (BRADSKI, 2000) is a free software library developed and maintained by Intel in 2000, to be a computer vision library focused on high performance and easy utilization, and it currently is on version 3.0. It is implemented on *C++*, has support for several languages, such as *Java*, *Python* and *Objective-C*, which helps the utilization of this library by many applications.

Mahotas (COELHO, 2013) is an image processing library for *Python*, having an implementation of more than 100 image processing algorithms, and most of its algorithms is implemented on *C++*, in order to have a high performance as well.

NumPy (OLIPHANT et al., 2006) is focused on the *N-dimentional* matrix representation and manipulation with high performance. It is used as a core to many other libraries such as *OpenCV*, because it is simple and very efficient.

Scikit-learn (PEDREGOSA et al., 2011) is an open source library heavily used on *machine learning*, it contains many predictive model implementations, such as *KNN*,

SVM, *Perceptron* and many more, besides preprocessing algorithms, such as *RFE*, *PCA*, normalization and more. Currently it is on version 0.19.2.

Keras (CHOLLET et al., 2015) is a high level neural network API, that focuses on an easy utilization of many deep learning methods. It also contains methods of feature extraction, by using distinct CNN architecture implementations, with ImageNet pretrained weighs as well.

3 STATE OF THE ART

In this chapter, selected works are presented which use *Digital Image Processing* and *Machine Learning* applied to blood cell images and Acute Lymphoblastic Leukemia (ALL) such as this work. Due to the fact that there are three main sorts of applications with this specific topic, which are: *segmentation*, *counting of blood cells* and *classification*, this chapter is divided into three sections, one for each topic.

3.1 SEGMENTATION PROPOSALS

3.1.1 Robust Segmentation and Measurement Techniques of White Cells in Blood Microscope Images

In (SCOTTI, 2006), the benefits of a system capable of segmenting white blood cells were initially discussed, once the author argues that transmitting digital images obtained by using a digital camera coupled to a microscope is more efficient than transporting blood samples to a laboratory to be analyzed. Also the operator that analyzes blood slices in order to detect and count white blood cells can suffer from tiredness, thus slowing down the process.

As one of the first papers to work with white blood cell segmentation, the author designs a complete algorithm to detect leukemia in peripheral blood images (Figure 21), but in this paper his focus was on the initial part of the system, which was to correct the lack of illumination, propose a measure to quantify if a method applied on this problem is performing well, segmenting white blood cells and making the further proposed steps that involve classification viable (SCOTTI, 2006).

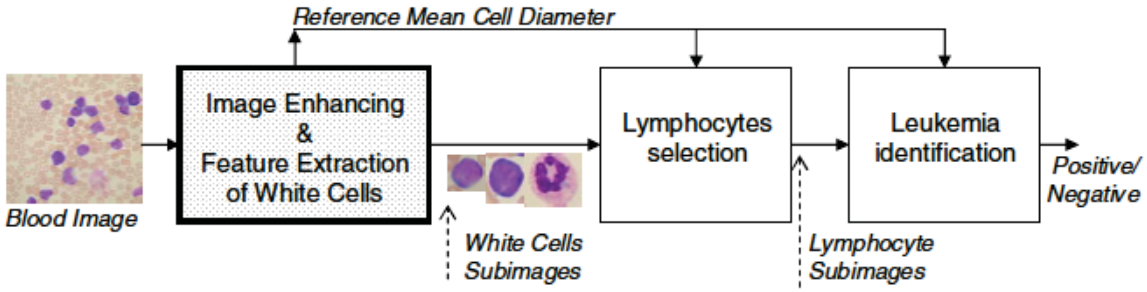


FIGURE 21 – Modules of an automatic system for acute leukemia detection, the paper proposed implements only the gray box phase. **Source**: (SCOTTI, 2006).

The first problem found when starting to work with images acquired by using a digital camera coupled to a microscope was the irregular illumination, thus compromising the effectiveness of simple threshold methods on background removal. Two solutions to correct the illumination were proposed, where the first one consists of applying a Gaussian

low pass filter on a grayscale converted image, then subtracting from the original one, while the other solution works with a group of at least twenty to forty images, so the average image can be computed before the application of the Gaussian low pass filter which generates a better result. After correcting the illumination problem, it was observed that the histogram became bi-modal, so a threshold using the Otsu's method split the background from cell pixels (SCOTTI, 2006).

The white blood cell segmentation method was based on the L*a*b color representation model, using a clusterization with three clusters and values from channels a and b, because these channels contain color representations and the objective was to differ white blood cells which tend to have more blue in them than other cells. This was the first paper that used color as a discriminator among blood cells, thus demonstrating its potential.

As one of the first works on this topic was needed to define a form of quantifying the results, a specialist marked the images manually, then the area of the markings and of the segmented white blood cells area were compared, and the results of the method and of the specialist markings correspond to 92% on the 243 cells tested.

3.1.2 Segmentation of Microscopic Images for Counting Leukocytes

(GAO et al., 2008) proposes a method focused on segmenting leukocytes to further count, once most of blood diseases diagnoses is made based on blood cells counting. Their focus was to develop a method with high performance and cytoplasm preservation, while other works focused only on leukocyte nucleus segmentation, due to difficulties in leukocyte cytoplasm segmentation, such as instability of staining.

The proposed method uses a non-decimated transformed complex wavelet to extract a textural gradient, which was subject to a watershed algorithm and finally an adaptive threshold, in order to segment the leukocyte. This pipeline was tested on four color channels, i.e., *Saturation (HSI model)*, *b (L*a*b model), Intensity (HSI model)* and *Green (RGB model)*, but the *Saturation* channel was chosen as it presents the best results.

The authors tested a total of thirteen images of a private database, and according to the authors, the results were very consistent, concluding that this method can be used as a real time leukocyte segmentation method.

3.1.3 Segmentation of Leukocytes and Erythrocytes in Blood Smear Images

This paper presents a method focused on segmenting both leukocytes and erythrocytes as the majority of blood disorders affects the population of these cells in a patient. Techniques of *digital image processing* and *machine learning* were combined, in order to perform the segmentation (BERGEN et al., 2008).

Erythrocytes detection was made by using template matching. Erythrocytes tend

to have a singular shape, since there is only one kind of red blood cell, which is different from leukocytes that have sub-classes with distinct shapes. Leukocyte segmentation was made by using a Naive Bayes classifier, and trained using color features extracted from the *HSV* model.

Distinct from other works, the evaluation of the proposed method was made based on the standard deviation between several masks made by a group of specialists and the mask found by the method. This approach was performed in order to make the evaluation more robust, since different specialists can perform distinct masks for the same image (small differences).

The final results were made based on dice coefficient as this indicates the similarity of two samples. In this case the specialist and the method masks, reached a dice coefficient of 0.941 for the nucleus and of 0.957 for the complete leukocyte. As there weren't any specialist masks for the erythrocytes, an evaluation of erythrocyte segmentation using dice coefficient wasn't made either.

3.1.4 Segmentation of Leukocytes in Blood Smear Images Using Color Processing Mechanism Inspired by the Visual System

Different from the majority of the methods proposed for segmenting white blood cells that use classical *digital image processing* techniques in the segmentation process, (WU et al., 2009) propose an approach inspired by the human eye biology. While the human eye uses three types of cones to interpret colors and rod brightness, Wu designed an *Artificial Neural Network*, where its input layer receives a feature vector with four features, i.e., Blue pixel value, Green pixel value, Red pixel value and N that represents the brightness, illustrated in Figure 22.

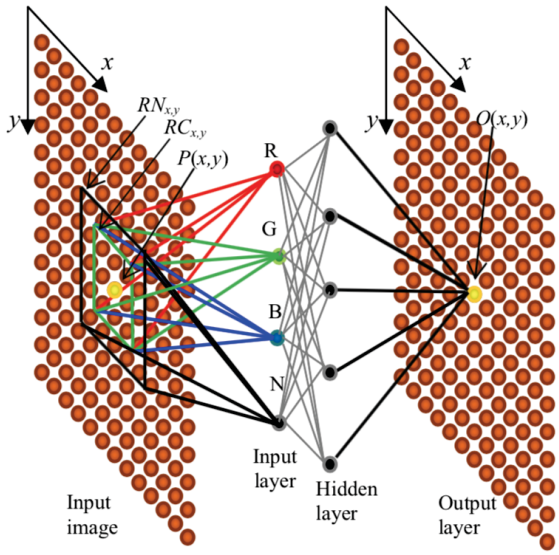


FIGURE 22 – Architecture of Neural Network Model. **Source**: (WU et al., 2009).

All pixels of the image that will be segmented must be classified among four classes (background, erythrocytes, leukocyte nucleus and leukocyte cytoplasm) generating a mask for each class. The data used to train the ANN was selected manually from images used in the experiments, so there is not a fully automatic procedure. The authors comment that the results were good, but they do not give any measurement data and neither do they tell how many images were tested. Even if the idea was very interesting, the lack of results makes it difficult to compare them with other works.

3.1.5 A Leukocyte Nucleus Segmentation Scheme Based on Fingerprint Smoothing

As the most discriminant characteristic used on leukocyte class differentiation is its nucleus shape, a method focused on segmenting only leukocyte nucleus was elaborated by (HAO et al., 2010b).

The authors observed that the G color channel (*RGB model*) tends to have three distributions, but finding an optimal value that splits the distributions was a hard task on the original histogram, and in order to solve this problem an approach based on fingerprint smoothing was applied, which rearranged the histogram, making the distributions much easier to detect and split.

After the threshold application, there were still some holes in the binary nucleus, which were closed without considerable information loss using mathematical morphology. The authors tested this method on 120 images containing only one leukocyte from a private base provided by the Department of Hematology of the 3rd Hospital of Qinhuangdao, and on all images the tests were successful where leukocyte nucleus were segmented without considerable loss. In order to confirm the method generalization, four more images were tested with distinct characteristics, these being the existence of more than one leukocyte and distinct illumination patterns in one image. The result of this second test shows that this method doesn't suffer any impact when facing these conditions.

3.1.6 An Adaptive Leukocyte Nucleus Segmentation Using Genetic Algorithm

(HUANG et al., 2012) propose a segmentation method focused on segmenting only the leukocyte nucleus, mainly because the quality of the stained cytoplasm varies according to temperature, concentration and reaction time, so it is possible that blood smear images produced in the same laboratory by the same biologist were not identical.

Same objective as (HAO et al., 2010b), but with an approach based on combining channels of distinct color representation models to enhance leukocyte nucleus, in order to isolate it in further steps. The paper showed that the G channel (*RGB model*) represents the leukocyte nucleus with a bigger contrast than other color channels, but it suffers with light conditions, so the Saturation channel (*HSV model*) was chosen to be combined with the G channel, once it is insensitive to light conditions.

After nucleus enhancement, a genetic algorithm was applied in order to detect optimal threshold value using statistical characteristics of the combined image, as adaptive threshold and Otsu's method don't perform well, according to the authors. The evaluation was done using dice coefficient (DC) and relative distance error (RDE), reaching 0.977 on DC and 17.6 on RDE.

3.1.7 Robust Leukocyte Segmentation in Blood Microscopic Images Based on Intuitionistic Fuzzy Divergence

(DANYALI et al., 2015) propose a method of leukocyte segmentation combining channels from distinct color representation models. The authors used fuzzy divergence on channel *a* (model L*a*b) to segment the leukocyte nucleus, and extract the background by applying Zack's algorithm (ZACK; ROGERS, 1977) on channel *M* (model CMYK). Since it has a big contrast between background and other elements, these steps were used as a starting point to a more robust pipeline.

After extracting leukocyte nucleus and removing background, the next step focused on recovering the leukocyte cytoplasm then removing the erythrocytes attached to them, using watershed algorithm. The authors proved the robustness of the method, by testing in three databases with distinct acquisition methods and illumination patterns. On the IDB1 subset composed of 33 images, it reaches an accuracy of 98%.

3.1.8 Unsupervised Leukemia Cells Segmentation Based on Multi-space Color Channels

As seen in other papers, (VOGADO et al., 2016) also use channels from different color representation models on leukocyte segmentation, but the differential on their approach is that it uses the *K-means* clusterization algorithm and morphological operations on its pipeline.

The pipeline starts by applying a *median filter 7x7* on channels *M (CMYK model)* and *b (L*a*b model)*, then subtracting the resulting images. The following step uses clusterization algorithm *K-means*, by using the *K* parameter as 3, once there were three classes on the resulting image, i.e., leukocyte nucleus, leukocyte cytoplasm and background. It was noticed that after the subtraction, maintaining negative values generates a better result on clusterization. In the final step, an erosion morphological operation was applied then the regions with an area smaller than 800 pixels were removed as these regions were interpreted as noises.

As in (DANYALI et al., 2015), the proposed method was tested in three databases to verify its generalization, and the results presented were very solid, reaching a *Kappa Index* of 0.9342 on ALL_IDB2, 0.8603 on BloodSeg and 0.9119 in the Leukocyte databases.

3.1.9 Segmentation of White Blood Cell from Acute Lymphoblastic Leukemia Using Dual-Threshold Method

(LI et al., 2016) propose a segmentation method based on the threshold application on distinct color representation channels with different values, for the results to be combined afterwards. The authors divided the pipeline into three fundamental steps, as illustrated in Figure 23.

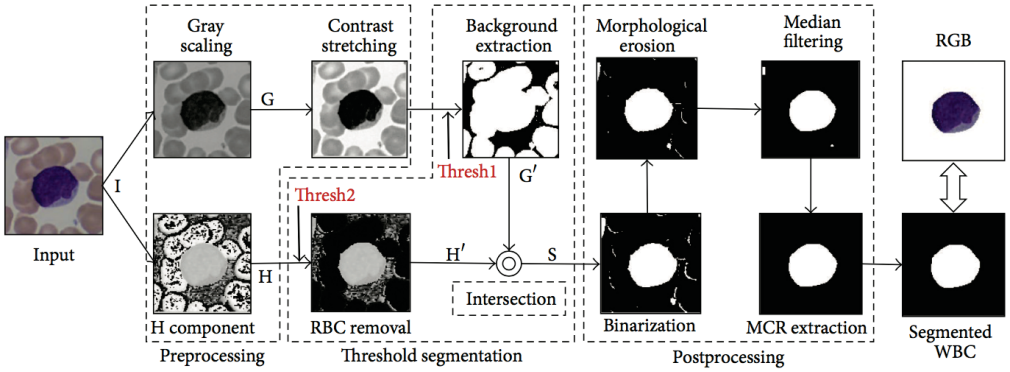


FIGURE 23 – Flowchart of the proposed dual-threshold segmentation scheme. **Source**: (LI et al., 2016).

The first phase aims at converting the *RGB* input image into a grayscale image, then increase its contrast, and converting the RGB input image into the *HSV* model and isolating the *Hue* channel. Optimal values for parameter T on both threshold operations were found by using the *Golden-section search method*, after the threshold operation resulting images were combined by an intersection operation (AND). Finally, the resulting image was binarized, making it possible to use an erosion operation, after eroding the image pass through a 15x15 medial filter, then extracting a maximum connected region, in order to remove small holes, finishing the segmentation pipeline.

The tests used a subset containing the first half of the ALL_IDB2 database (130 images). Two doctors were invited to manually segment all tested images, making an evaluation of the method possible. Dice Similarity Coefficient (DSC) was used to measure the method's efficiency, reaching a 0.9542 mean with a standard deviation of 0.04. Even the last 130 images of ALL_IDB2 are not included in the result. The authors commented that the method had great results too.

3.2 LEUKOCYTE COUNT PROPOSALS

3.2.1 A Novel Auto-Segmentation Scheme for Colored Leukocyte Images

As blood cells counting is used as the grounding on blood diseases diagnose, (HAO et al., 2010a) propose a novel method to segment and count leukocytes on colored blood images. Their method uses the *HSI model*, due to the fact that this color representation model has a much smaller correlation between its channels than RGB model. Channel

Saturation was chosen because it doesn't suffer with illumination changes, and the leukocyte nucleus tends to have a bigger saturation, due to its chemical affinity to the dye.

The segmentation phase starts by applying a *median filter* on the original image and then converting it to HSI and isolating the S channel. The saturation image was binarized using a threshold with a fixed value of 90. This value was found based on tests during the development, by segmenting the leukocyte nucleus. Finally, morphological operations of erosion and dilation were applied, in order to remove small objects interpreted as noises.

After the leukocyte nucleus was segmented, remaining objects were counted, resulting in the leukocyte image final count. The authors took only the leukocyte nucleus into consideration, since leukocyte is the only human blood cell that possesses a nucleus. From the 149 samples tested, 91.95% of the images has all leukocytes nucleus segmented correctly, resulting on the correct count.

3.2.2 Leukocyte Nucleus Segmentation and Recognition in Color Blood-smear Images

(HUANG; HUNG, 2012) propose a method of leukocyte automatic counting using *digital image processing* and *machine learning* algorithms, but for the majority of the proposed works focused on leukocyte counting the authors made the count of each leukocyte lineage individually.

The segmentation phase combines channels *Green* and *Saturation* from models *RGB* and *HSI*, then applied multiple threshold operations using the Otsu's method. The final segmentation step consisted of removing the noise regions based on the erythrocytes area. If the region was smaller or much bigger than the mean erythrocyte area, the region was classified as noise and removed.

At the end of the segmentation phase, a total of 85 features were extracted, from which 80 texture features were extracted using GLCM and 5 shape features. As 85 features would demand a great computational power in the classification phase, the authors used *Principal Component Analysis (PCA)* to reduce its dimensionality from 85 to 7, thus reducing the size of the feature vector, but almost maintaining the representation.

The classification was made using the *K-means algorithm*, where the *K (number of clusters)* parameter is determined by a genetic algorithm, resulting on the best suited value on each case. The results were very consistent reaching a dice coefficient of 0.95 in the segmentation phase, and the classification phase reached an accuracy of 90.98%.

3.2.3 An Automated Framework for Counting Lymphocytes From Microscopic Images

A framework focused on lymphocyte counting with low computational and equipment cost was proposed by (LE et al., 2015), where the authors limited themselves to

using only low cost *digital image processing* algorithms, thus ensuring a low computational cost. Even through it was focused on low cost, the framework presented very consistent results, reaching a 90% accuracy in the ALL_IDB1 database, a database known as very challenging by having irregular illumination and zoom.

Distinct from many other works that used one or more channels from *RGB*, *L*a*b* or *HSI* color representation models, (LE et al., 2015) used the *Haematoxylin-Eosin-DAB (HED)* model, since it is credited as being the most suitable color representation model to be used in medical problems.

As illustrated in Figure 24, the HED representation of leukocytes made it very simple to segment them, and the remaining steps of the proposed framework focused on splitting the attached leukocytes, by using a bilateral filter followed by the *Canny edge detector* and finally, a watershed algorithm.

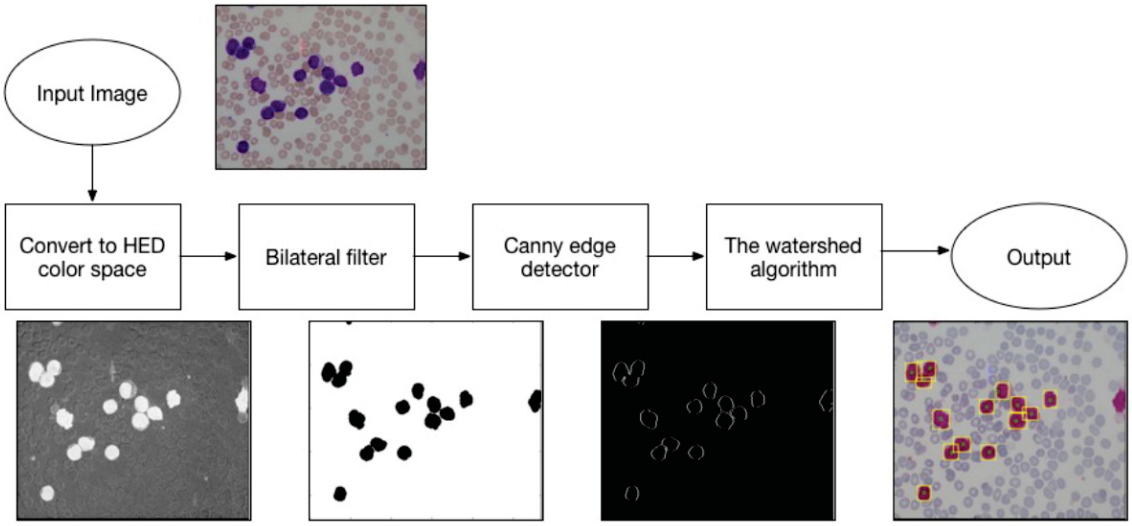


FIGURE 24 – The overview of the proposed method. **Source**: (LE et al., 2015).

3.2.4 Automated Leukaemia Detection Using Microscopic Images

(PATEL; MISHRA, 2015) propose a leukemia detection method. However, their current results focuses on the leukocyte counting, since this is the result presented. The preprocessing phase applied *median* and *wiener* filters to remove image noise, the leukocyte segmentation started by applying the *K-means* clustering algorithm on the *Green* channel (*RGB* model), because the authors noticed that the leukocyte nucleus tends to be the darkest region in this channel. So after the clustering conclusion, a histogram equalization followed by *Zack algorithm* was applied, resulting in the leukocyte nucleus segmentation.

Once an image containing only a leukocyte nucleus was obtained, a clean step was made to remove unwanted objects, such as incomplete leukocytes and attached leukocytes. At that point, the authors chose to discard the attached leukocytes and work only with

isolated leukocytes, but in future works they plan to design a slit method. The object removal was based on its *roundness* and *mean area*. After calculating the roundness of all remaining objects with a value lower than 0.80, it was interpreted as an attached leukocyte and then removed. Finally, the mean area was used to detect incomplete leukocytes, since they have a small area.

The tests were performed using a set of twenty seven images from the ALL_IDB1, which were selected for presenting the same lighting condition. On these images, 93.57% of the leukocytes were detected correctly.

3.2.5 A Computer-Aided System for Differential Count from Peripheral Blood Cell Images

(LODDO et al., 2016) propose a segmentation method based mostly on *machine learning* methods, after verifying that the *digital image processing* based algorithms proposed to segment blood smear images tend to suffer a huge impact when facing distinct lighting patterns and distinct staining methods, which is common on blood smear images, since distinct devices and procedures can be used for image acquisition.

Initially, a set of images were used to extract three different *Regions Of Interest (ROIs)* illustrated in Figure 25a, i.e., *background, erythrocyte, leukocyte* regions. These regions had their pixel values extracted from all the *RGB* channels to be used as feature vectors. Since there was a large amount of feature vectors, where a significant portion possibly had the same values, a *Nearest Neighbor Search (NNS)* was used to remove redundant data and outliers, thus decreasing the computational power needed to train a predictive model.

An *SVM classifier* with a *Radial Basis Function* kernel was trained, by using feature vectors extracted from the ROIs and used to classify all pixels on a blood smear image. These pixels were classified among *background, erythrocyte and leukocyte*, resulting on a mask capable of isolating each class individually, exemplified in Figure 25(b).

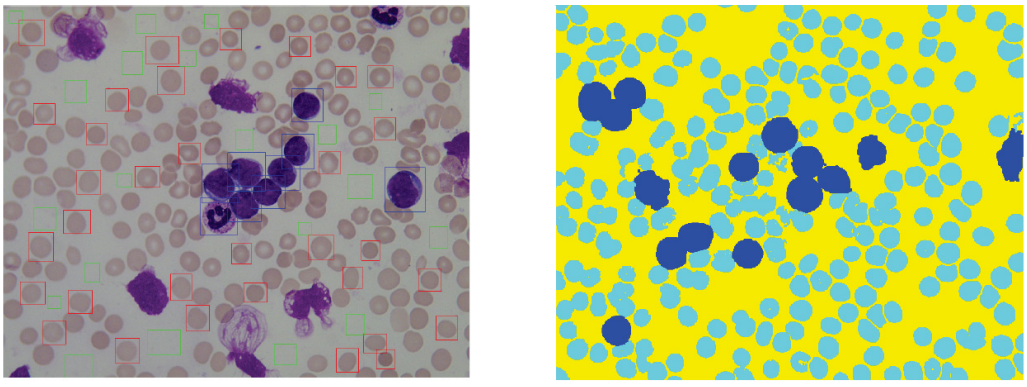


FIGURE 25 – (a) ROIs used in training phase, blue boxes refers to leukocytes, red erythrocytes and green to background, (b) mask resulting of SVM pixels classification. **Source**: (LODDO et al., 2016).

As some images may have leukocyte agglomerations and can not be ignored, *circular Hough transform* was used to detect how many leukocytes were present in an agglomeration, because leukocytes tend to have a circular shape. After classifying all image pixels and generating a mask that segmented only leukocytes, the resulting mask was applied on the original image, then it was converted to grayscale, making it possible to use a *circular Hough transform* on it.

The counting processes consist of adding the number of leukocytes that compose the agglomerations with the isolated leukocytes. Tests were done by training the*SVM* using the ROIs extracted from the ALL_IDB2 complete database, in order to segment and count a set of 33 ALL_IDB1 images, resulting in a 99.2% of white blood cell count and 98% of red blood cell accuracy count, outperforming the state-of-the-art, according to the authors.

3.3 CLASSIFICATION METHODOLOGIES

3.3.1 Computer Based Acute Leukemia Classification

(FARAG, 2003) proposes a computer based classifier system for the *acute leukemia* classification, able to distinguish *lymphoid* from *myeloid*. The author chose to use grayscale images, in order to reduce computational cost. A total of 401 spatial features were extracted from the grayscale images, where 200 come from the horizontal average, 200 others from the vertical average (images used had sizes of 200x200) and a complete image average was the last extracted feature.

Once again, aiming on decreasing the computational power needed, *Fisher criterion* was applied, in otder to select only the most important features, thus reducing the feature vector from 401 to 5. An *artificial neural network* with two hidden layers was used in the classification phase, where half of the fifty images from the *Faculty of Medicine at Cairo University*, were from *lymphoid leukemia* cases and the other half from *myeloid leukemia* cases. There was a 0.06% error through this method, which was a interesting result as one of the precursors showed spatial feature potential for this particular problem.

3.3.2 Morphological Classification of Blood Leukocytes by Microscope Images

(PIURI; SCOTTI, 2004) present a fully automated pipeline to segment and classify leukocytes, according to their lineage. The method works with grayscale images, since the colorant used to enhance white blood cells does not have the same effect, resulting in different pigment intensities.

The pipeline was divided into four phases, where the first one was focused on enhancing the input image, so that the next step was able to segment the leukocytes in a more effective way. After they were segmented, 23 morphological features used on the

leukocyte classification were extracted. A total of five *Artificial Neural Networks* were trained, where one ANN was specialized in a leukocyte lineage, so one leukocyte was classified by the five ANNs individually, and the one with the highest degree of certainty, determined the final class.

The experiments were made on a private dataset with 113 images, with a total of 134 leukocytes labeled by experts. The proposed parallel ANN architecture reached a mean error of 0.08 with a standard deviation of 0.09. As one of the precursors, the results proved the possibility of identifying leukocyte lineage based morphological features.

3.3.3 Leukocyte Segmentation and Classification in Blood-smear Images

(RAMOSER et al., 2005) propose a fully automated approach, in order to segment and classify leukocytes, according to its lineage from blood smear images. The authors opted to use the *HSL* model instead of RGB on the segmentation phase, since it is less sensitive to illumination variation according to the authors. The *K-means* clustering algorithm was used on segmentation pipeline like other segmentation proposals. The value attributed to the K parameter was three, as there were three distinct clusters in the image, i.e., background and leukocyte cytoplasm, erythrocytes and leukocyte nucleus.

After the clusterization, a probability image was built to segment only leukocytes, where this image had high values for pixels with big chances of belonging to a leukocyte, and low values if the pixels had higher chances of not belonging to a leukocyte. An adaptive threshold applied on the probability image was used to finish the segmentation phase.

Different from (FARAG, 2003) and (PIURI; SCOTTI, 2004) that used only one kind of feature to compose the feature vector (spatial and morphological), (RAMOSER et al., 2005) combined two kinds of features, i.e., eighteen color statistic features and eight shape features extracted only from the leukocyte nucleus, resulting in a feature vector with twenty-six features.

The classifier used was the *SVM*. Actually thirteen support vector machine classifiers were used, as each one was trained using the one-vs-all approach to be specialized in one of the thirteen possible classes. Classifications were made based on the probability resulting from each classifier, and the one with the highest certainty degree was chosen as the final class, but if none of the classifiers returned a probability higher than 50% the sample was rejected. This architecture showed a great result, reaching above 90% accuracy with a rejection on a set of 1.166 images.

3.3.4 Automatic Morphological Analysis for Acute Leukemia Identification in Peripheral Blood Microscope Images

(SCOTTI, 2005) was one of the first proposals focused on the *Acute Lymphoblastic Leukemia* detection, where the objective was to develop an efficient feature extraction and a classification method that distinguishes a leukocyte carrier of *ALL* from a healthy one, since the leukocyte segmentation pipeline used was that of (PIURI; SCOTTI, 2004).

Even with a pipeline that detected and extracted sub-images with each leukocyte in its center already developed, the authors had to design a pipeline capable of segmenting leukocytes in the center of each sub-image, and then segment the leukocyte nucleus and cytoplasm, and be able to extract features from each of these regions individually.

The central leukocyte segmentation starts with a *Sobel edge enhancing*, improving the image condition for the application of the *Canny edge detection*. After that a morphological operation of dilation was applied in order to connect contours that still could be disconnected. After that the central contour was flooded to finally apply the morphological operations once again, but this time an erosion removing all the objects, except for the central leukocyte. Once the central leukocyte segmentation was completed an Otsu's threshold method was applied to segment leukocyte nucleus from cytoplasm, because their histogram tends to be a bi-modal which fits well in this method.

When the nucleus and the cytoplasm were segmented, a total of twenty-three features were extracted, so as to represent the leukocytes. An imageset with 113 images of complete blood smear provided by the M. Tettamanti Research Center for Childhood Leukemias and Hematological Diseases, Monza, Italy, was used to test the proposal. From the imageset, a subset with 150 images of isolated leukocytes was created with expert help, so that the classification tests would be done, where a mean error of 0.0113 with a standard deviation of 0.0281 was reached by a *feed forward neural network*.

3.3.5 New Decision Support Tool for Acute Lymphoblastic Leukemia Classification

(MADHUKAR SOS AGAIAN, 2012) made an analysis of several proposals focused on blood smear image segmentation and acute lymphoblastic leukemia detection, and concluded that all of them only extracted features from isolated leukocytes, so (MADHUKAR SOS AGAIAN, 2012) proposed that features were extracted from a complete blood smear image.

The pipeline was divided into three phases, starting on leukocyte nucleus segmentation using a *K-means* clustering algorithm on channels a and b, as these channels are responsible for the color representation on the L^*a^*b model.

Texture and shape features were extracted from the entire segmented image in order to generate a representation, and an *SVM* classifier was used to perform the

classification. A set of 98 images provided by Charles Fabio Scotti was used to validate the method, and a 93.5% accuracy was reached using a *K-ford cross-validation* method, in order to ensure that there wasn't overfitting.

3.3.6 Identification and Classification of Acute Leukemia Using Neural Network

(FATMA; SHARMA, 2014) propose a method of leukemia detection in blood smear images by using *Artificial Neural Networks*. Because it is focused on a high performance segmentation phase, it used only *digital image processing* algorithms that do not demand high computational power, in a way limiting their possibilities.

The segmentation phase consists of a contrast enhancement, followed by a *K-means* clustering isolating the majority of leukocyte nucleus pixels, and a median filter is also applied at the end to remove the remaining noise. As focused on high performance, and only a few features were extracted to decrease the computational power. The features chosen to build a leukocyte representation were radius, perimeter, mean, standard deviation, area, variance and circularity.

Even using simple *digital image processing* and *machine learning* algorithms, (FATMA; SHARMA, 2014) reached an accuracy rate of 91%. A subset of fifty images, selected from the *ALL_IDB1* database was used to perform the tests, and it should be emphasized that thirty-eight images were used to train the net, and the remaining twelve used for classification.

3.3.7 Leukocyte Classification for Leukemia Detection Using Image Processing Techniques

(PUTZU et al., 2014) propose a fully automated pipeline for the *ALL* detection in blood smear images, illustrated in Figure 26. Different from other works with the same objective, the authors do not discard agglomerated cells and also preserve leukocyte cytoplasm during segmentation, with it being considered more robust than the others.

The leukocyte segmentation was done by applying a threshold on *channel Y (CMYK model)*, where the T value was found by using the *Zack algorithm*. As major problems encountered on blood smear image classification are constituted presence of leukocyte agglomerations and incomplete leukocytes, a separation method was done by first identifying the agglomerations and isolating leukocytes based on their roundness. After that incomplete leukocytes were removed based on roundness measure.

Once leukocyte agglomerations were detected, the separation was done using the watershed method followed by morphological operations, as seen in other articles. An extra segmentation step was taken in order to extract some features from the complete leukocyte (nucleus and cytoplasm) and others only using leukocyte nucleus. In order to do so, the *Zack algorithm* was used again, but this time applied on the *Green channel*, and as

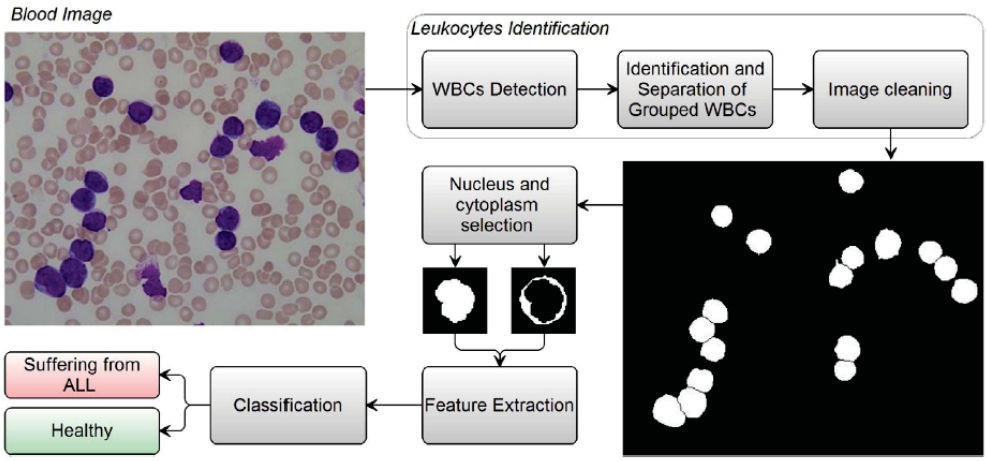


FIGURE 26 – Diagram of the proposed method from blood image to the ALL classification via identification of white blood cells (WBCs). **Source:** (PUTZU et al., 2014).

observed in other articles, this channel tends to highlight the leukocyte nucleus.

When all leukocytes from the input image were segmented and isolated, a total of 131 features were extracted to represent a leukocyte, where 30 were shape descriptors, 21 color descriptors and 80 texture descriptors. A group of classifiers was tested in order to find which one had the best result, so it could be chosen to be the definitive one, and out of *SVM*, *KNN*, *Naive Bayes* and *Decision Tree*, *SVM* was the one with the best result, reaching an accuracy of 93% and a sensitivity of 98% on a set of 33 images with the same lighting pattern selected from ALL_IDB1.

3.3.8 An Intelligent Decision Support System for Leukemia Diagnosis Using Microscopic Blood Images

(NEOH et al., 2015) used a clustering algorithm on the *L channel (L*a*b model)* to segment leukocytes. Different from other proposals that use the *K-means algorithm* in order to cluster the image, (NEOH et al., 2015) proposed a new clustering algorithm that was based on *simulating discriminant measure (SDM)* with in and between cluster scatter variances, resulting in a better performance than *K-means algorithm*, once it bases itself only on cluster scattering.

As the proposed clustering method successfully segmented the complete leukocyte, a consistent representation was extracted using color, shape, texture and statistical features. A feature vector with eighty features was elaborated to represent leukocytes. The authors opted not to use many features, since there wasn't a large amount of data to train the classifiers, thus avoiding the curse of dimensionality, as a set of 180 images from ALL_IDB2 was used to validate their method.

Different from other proposals which also tested several classifiers and concluded

that the *SVM* method was the best one to be used on blood smear image classification, due to its results, (NEOH et al., 2015) found its best accuracy result using the *Dempster-Shafer* ensemble method, reaching 96.72%, while *SVM* reached a 96.67% accuracy.

3.3.9 Naive Bayesian Classifier for Acute Lymphocytic Leukemia Detection

(SELVARAJ; KANAKARAJ, 2015) propose a simple method to segment and classify leukocytes as carrier of ALL or not, using the *K-means clustering algorithm* to segment the leukocyte nucleus, and a *Naive Bayes classifier* to define if the leukocyte has ALL or not.

The authors used only a set of forty images selected from the ALL_IDB2 database to perform the tests, where twenty images had leukocytes with ALL and the remaining twenty were from healthy patients. As the authors used only features extracted from the leukocyte nucleus, the application of a *K-means algorithm* provided a good nucleus segmentation, as seen on many other works.

After the segmentation, a representation with eight features was extracted from the leukocytes. According to the authors, a small representation was decided as a result of using a small dataset. Also considered a simple classifier, *Naive Bayes* presented a considerable accuracy, reaching 75%. So it can be concluded that by also using a simple pipeline with basic *digital image processing* and *machine learning* algorithms the result was expressive, due to its simplicity.

3.3.10 Leukocyte Classification in Microscopy Images for Acute Lymphoblastic Leukemia Identification

The proposal of (RODRIGUES et al., 2016) was focused on evaluating multiple linear classification models on the ALL detection problem, so the segmentation step was not as refined as seen in other proposals, which can be directly related to the 85% best accuracy found. The segmentation step started from a median filter on the *V channel (HSV model)*, followed by a binarization using Otsu's method, and finally morphological operations in order to close small holes of the segmented object.

A total of seven features related to shape were extracted, in order to build a leukocyte representation. Multiple predictive models were tested to perform the classification between positive and negative, i.e., *Decision Tree*, *Naive Bayes*, *KNN* and *SVM*, where *KNN* was the one with the best accuracy, reaching 85%, and *Naive Bayes* was the a worse with 70.3% accuracy.

Also presenting a relatively low accuracy in comparison to other articles that also used ALL_IDB2 to evaluate their method, (RODRIGUES et al., 2016) was one of the few that used all ALL_IDB2 images, and not only a subset with the same lighting. So

they faced many more difficulties than the proposals that used only small partitions of the original database, making this a more relevant result.

3.3.11 A Leukemia Diagnostic System Using Pre-Trained CNN's and a Classification Committee

(VOGADO et al., 2017b) proposed an approach using multiple pre-trained *Convolutional Neural Networks (CNNs)* to perform feature extraction. The authors did not use a segmentation step either as seen in many other articles, and stressed that a segmentation step resulted in worse performance of feature extraction via CNNs.

Due to the huge amount of features generated by CNNs, a total of 4.096 features, instead of only 108 samples of the complete ALL_IDB1 database, the *Principle Component Analysis (PCA)* feature selection method was used to reduce data dimensionality.

The classification was made using a classifier committee composed of *SVM, MLP and Random Forest*, combined with the majority vote technique. The combination of distinct classifiers resulted in an accuracy of 98.14%, being the best result found in literature.

3.4 DISCUSSION

After analyzing several proposals which aimed to *segment*, *count* and/or *classify* the blood smear images using *digital image processing* and/or *machine learning* methods, some tendencies could be observed indicating their efficiency.

In the segmentation phase, there was a considerable number of methods that used the *Green channel (RGB model)* in the leukocyte nucleus segmentation. Also, in the segmentation phase, it was noticed that color models *HSV* and *CMYK* had a real potential in this particular problem.

It was also observed that morphological and texture features tend to generate great results, but the most current work analyzed revealed that CNNs generate great results as well. *SVM* was also unanimously considered as the best classifier to be used alone. Table 2 presents a summary of the proposals focused on ALL detection or leukocyte classification as carrier of ALL or not. It was observed that a small number of proposals used the complete ALL_IDB database in their tests without any explanation as to why the subset was used, so their results can be considered inconclusive when applied to a small amount of images.

TABLE 2 – Resume of classification proposals

Authros	Database used	Number of images tested	Classification objective	Results
[Farag, 2003]	Private	50	Classify leukocytes as carriers of lymphoid or myeloid leukemia	Error of 0.06%
[Piuri and Scotti, 2004]	Private	113	Identify a leukocyte lineage based on morphological features	Mean error of 0.08
[Ramoser et al., 2005]	Private	1166	Identify a leukocyte lineage based on color statistics and nucleus morphological features	Accuracy above 90%
[Scotti, 2005]	Private	150	Detect the presence of ALL on isolated leukocytes.	Mean error of 0.0133
[Monica Madhukar, 2012]	ALL_IDB1	98	Detect presence of ALL, but extract features from the complete image	Accuracy of 93.5%
[Fatma and Sharma, 2014]	ALL_IDB1	50	Detect presence of ALL on blood smear images using low cost algorithms	Accuracy of 91%
[Putzu et al., 2014]	ALL_IDB1	33	Detect the presence of ALL on blood smear images, doesn't discard leukocyte agglomerations, and tested multiple classifiers.	Accuracy of 93% and Sensitivity of 98%
[Neoh et al., 2015]	ALL_IDB2	180	Uses a novel clustering method to segment complete leukocytes with the objective of ALL detection.	Accuracy of 96.72%
[Rodrigues et al., 2016]	ALL_IDB2	260	Evaluate multiple classifiers on complete ALL_IDB2	Accuracy of 85%
[Vogado et al., 2017]	ALL_IDB1	108	Uses CNN's as a feature extractor, after reduce the number of features using PCA a committee of three classifiers were used to detect ALL	Accuracy of 98.15%

4 METHODOLOGY

In this chapter, we present the image database used in the development of this work, as well as a detailed description of each step of the method proposed in order to to classify peripheral blood images.

4.1 IMAGE DATABASES

The image databases ALL_IDB1 and ALL_IDB2, both provided by Labati et. al. (2011), were chosen to be used in the development of this work, since they are free, public domain, labeled and developed for the purpose of testing digital image processing and pattern recognition algorithms. The image names on both bases follows the pattern ImXXX_Y.tiff, where XXX is an integer sequence used as an identifier of each image, and Y is the label that defines if the image belongs to a healthy patient, or a patient with ALL(if Y = 0 the patient is healthy, and if Y = 1 the patient as ALL). The images were captured using a Canon PowerShot G5 camera attached to a microscope. It is also important to emphasize that all ALL_IDB2 images are cutouts of ALL_IDB1 images(LABATI et al., 2011).

4.1.1 ALL_IDB1 Database

The ALL_IDB1 database consists of 108 images taken from blood samples collected and labeled by medical oncologists in September 2005, where 49 images belong to patients with ALL and the remaining 59 images belong to healthy patients. Only 510 of the approximately 39.000 blood cells existing in all images were labeled as lymphoblasts, which shows the need to remove the existing erythrocytes and thrombocytes before classifying an image. Two important characteristics verified were different kinds of lighting and zoom level during acquisition, as we can see in Figure 27 .

4.1.2 ALL_IDB2 Database

The ALL_IDB2 base contains 260 TIFF-format images, equally divided between diseased cells, called lymphoblasts, and healthy cells, derived from cutouts of some ALL_IDB1 base images. The images were composed of one central leukocyte (the object for classification) and some erythrocytes and/or leukocytes, that must be removed. Figure 28 shows examples of images from ALL_IDB2.

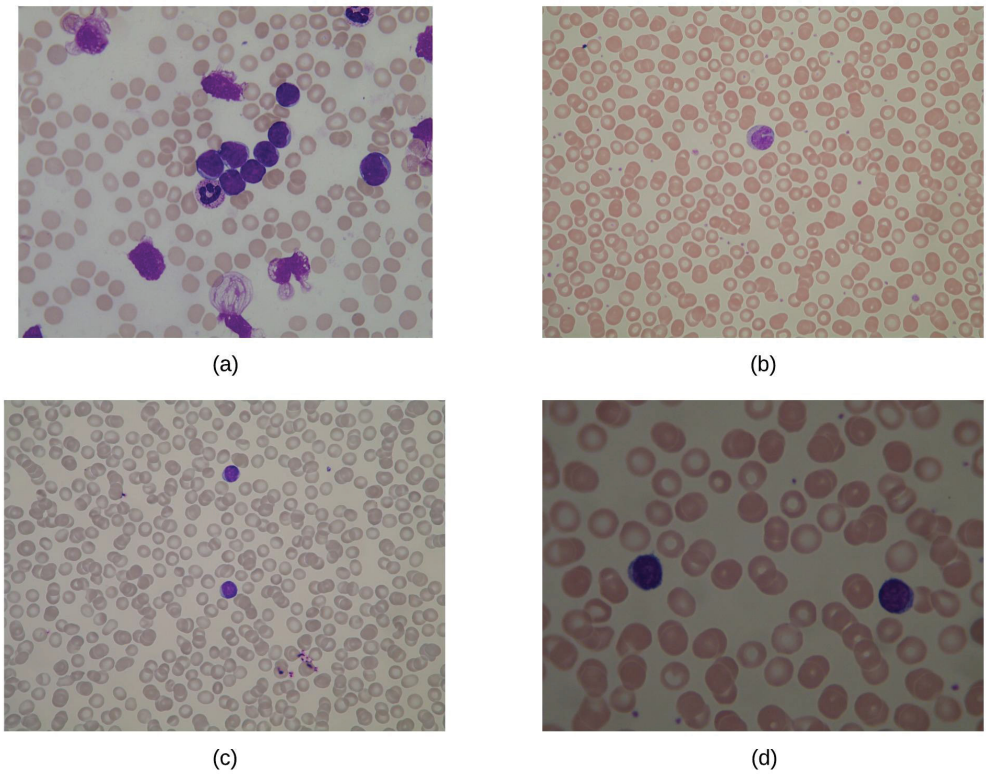


FIGURE 27 – Images that show many different kinds of lighting and the zoom level present on the ALL_IDB1 base. **Source**: (LABATI et al., 2011)

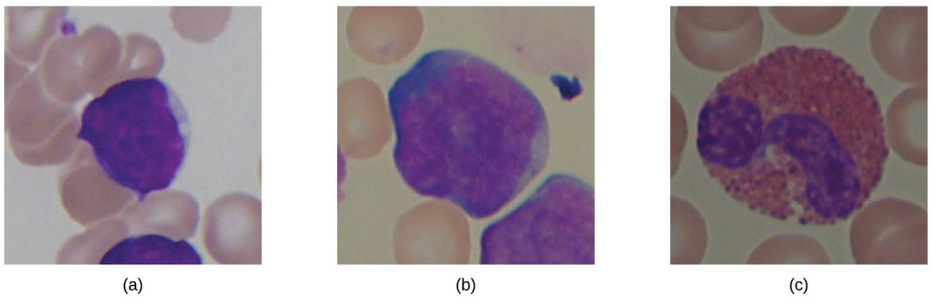


FIGURE 28 – ALL_IDB2 image examples that show the existence of erythrocytes and incomplete leukocytes on the extremities, and one leukocyte in the center. **Source**: (LABATI et al., 2011)

4.2 OVERVIEW

Our approach is to classify an ALL_IDB1 image based on local regions of importance. Different from other works that classify these images using global information, extracting features from a complete ALL_IDB1 image, not taking into account local regions of importance.The proposed work can be divided into five main steps. As can be seen in Figure 29, the first one focuses on segmenting the existing leukocytes in an ALL_IDB1 image. Thereby, the second step was to find the center of mass of each leukocyte and

cutout sub-images from the original ALL_IDB1 image, using these centers of mass found as central coordinates for the sub-images, reassembling the ALL_IDB2 image pattern, as illustrated in Figure 30. This step was important because the ALL_IDB2 images were used to train the classifier.

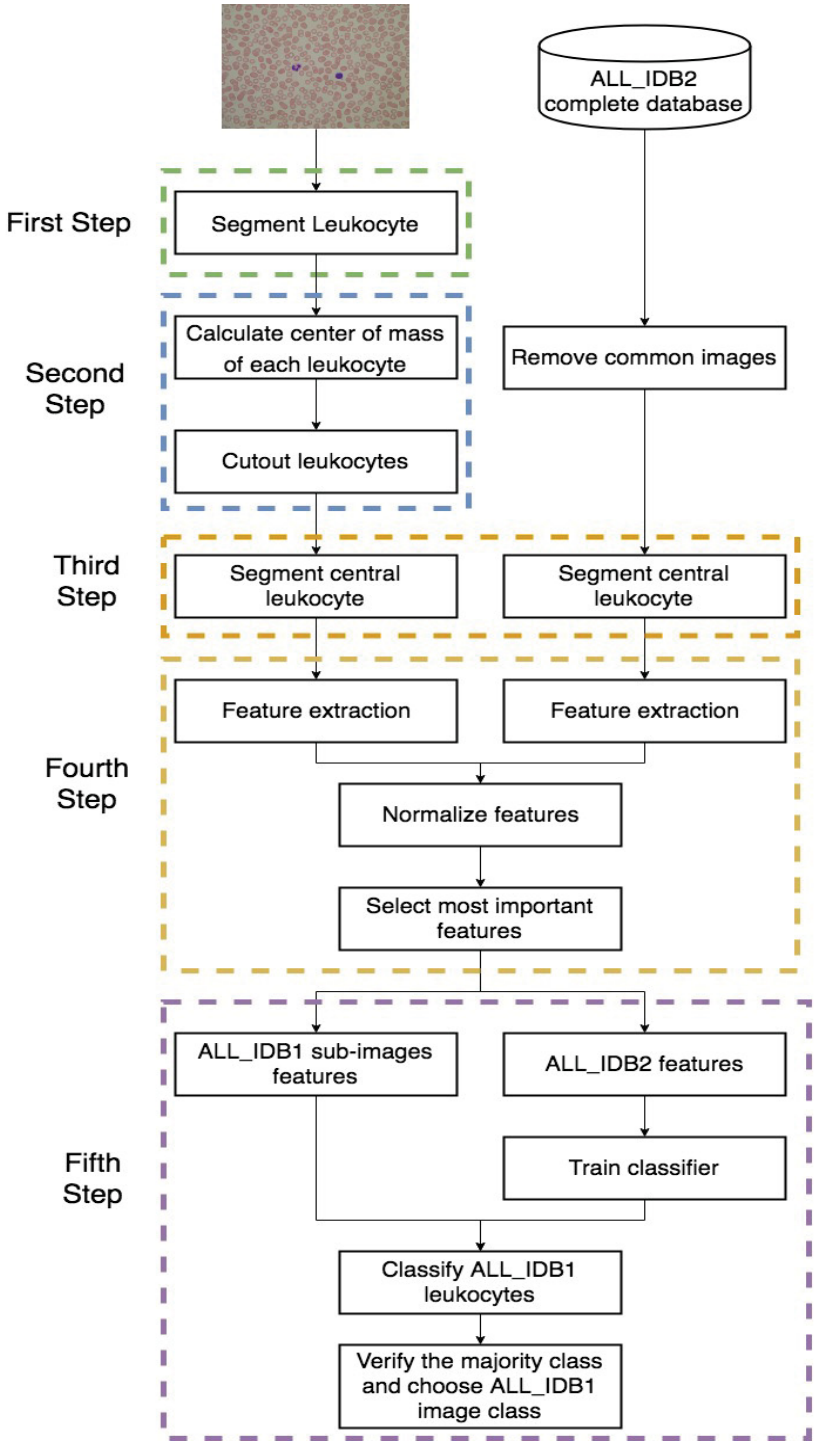


FIGURE 29 – Complete pipeline proposed, where each colorful dash represents one step described in the previous section, where the green dashes represent the first step, the blue ones the second step, the orange ones the third step, the yellow ones the fourth step and the purple ones the fifth step. **Source**: the author.

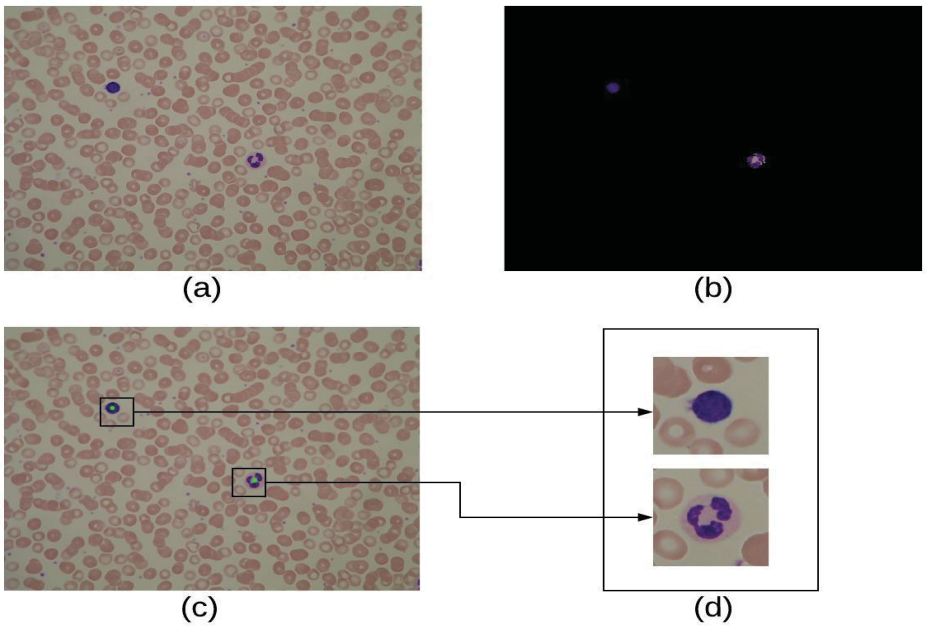


FIGURE 30 – This figure exemplifies the steps used in leukocyte extraction. Where (a) is the ALL_IDB1 image, (b) is the result of the pipeline that segmented all leukocytes existing on (a), (c) calculates the center of mass of each leukocyte, represented by a green dot, (d) represents the cutout images resulting from this pipeline. **Source**: the author.

The third step was focused on segmenting only the central leukocyte in an image. The pipeline was applied on sub-images resulting from the last step and the ALL_IDB2 images before feature extraction. Some feature extract methods used on the classification process work better if only the object of interest exists in the image.

The fourth step consists in extracting and working the features that were used in image classification. This step initiates by extracting features from sub-images, resulting from the third step, using the *Convolutional Neural Network VGG19*, and extracting texture features from segmented sub-images. Finally, it was binarized, normalized and reduced the dimensionality by selecting the most important features.

The fifth and final step consisted in training an *SVM* classifier by using features extracted from the ALL_IDB2 images and classifying each sub-image extracted from the ALL_IDB1 image. Lastly, the ALL_IDB1 image class was defined based on the majority class of its individual leukocytes.

4.3 DEVELOPMENT BASE

To ensure that any pipeline used in this work didn't have a bias that compromises its generalization power, the ALL_IDB1 database was divided in to two subsets, where 76 images, corresponding to approximately 70% of the complete dataset were used in

the development of the pipelines, always testing a new hypothesis on these images. The remaining 32 images, about 30% of the complete dataset, were used only when the proposal was finished, verifying that the proposed method didn't perform well only on the images used in its development process.

The 76 images selected to compose the subset used during the development process were selected based on the five lighting patterns observed in the ALL_IDB1 database. A proportional number of images was selected for each lighting pattern, ensuring that the subset used in the development represents the complete base well.

4.4 REDUNDANT ALL_IDB2 IMAGES REMOVAL

As there weren't correlations in the (LABATI et al., 2011) article, as to which ALL_IDB1 images were used to extract which ALL_IDB2 images, a technique was developed to confirm if an ALL_IDB2 image belongs to an ALL_IDB1 image. This technique consists in convolving an ALL_IDB2 image into an ALL_IDB1 image. Each convolution window was converted to gray scale Binarise, them calculated the *Pearson correlation coefficient* and finally verified if the result is bigger than 0.9, indicating a high correlation between the ALL_IDB2 image and this ALL_IDB1 area. This technique was applied in the entire ALL_IDB2 database images, and the results were stored and used when it was necessary to remove the ALL_IDB2 images in the classifier training.

4.5 ALL_IDB1 LEUKOCYTE SEGMENTATION

The directly segmented leukocytes of an image show themselves as a complex process, due to the existence of several elements that must be removed before some feature extraction, such as erythrocytes, the background itself and the incomplete leukocytes. The approach used was to individually segment each element, then remove them from the original image, remaining only leukocytes. In the next subsections each step developed is described.

4.5.1 Background Removal

It was necessary to divide the background segmentation pipeline into two main phases, where the first one is a preprocessing phase with the objective to correct the illumination irregularities and increase the difference from the background pixels of those belonging to leukocytes and erythrocytes, using Algorithm 3. Figure 31 shows the algorithm pipeline. The illumination problems are caused by the fact that the center tends to be illuminated with greater intensity than the rest of the blood slice, thus making the peripheral regions darker than the central region in some images.

Algorithm 3 Preprocessing algorithm:

1: Convert image from BGR to L*a*b color space
2: Isolate L channel
3: Blur L channel using a kernel (161, 161)
4: Invert pixel values, inside the range 0 to 255
5: Sum inverted image to original L channel image
6: Equalize added image
7: Reassemble L*a*b image using the equalized image as new value to L channel
8: Convert new L*a*b image to BGR color space

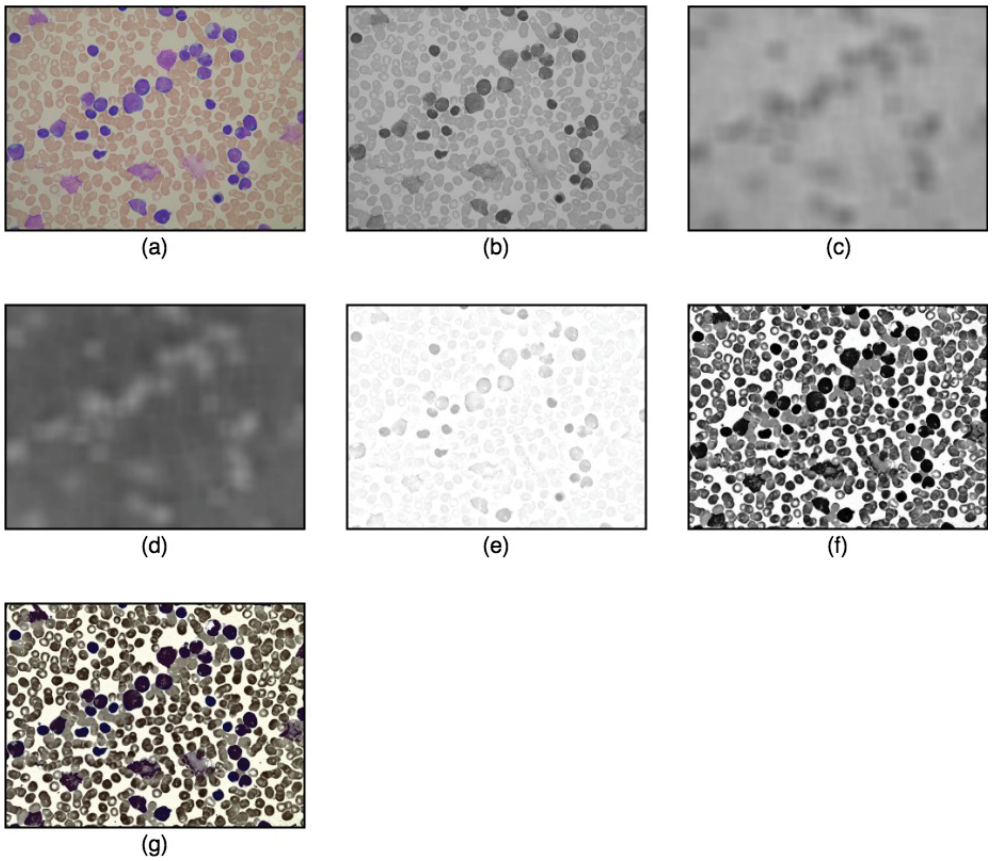


FIGURE 31 – Displays the result of each step of the algorithm 3, (a) original image, (b) L channel, (c) L blurred, (d) inverted image, (e) addition of (d) and (b), (f) equalized image, (g) preprocessed image. **Source**: the author.

The second phase used the preprocessing final image and focused on the segment background pixels, combining the blue and green color channels (RGB model) because these channels showed complementary characteristics. The blue channel tends to represent erythrocytes with low values, leukocytes with intermediate values and background with high values, while the green channel represents erythrocytes with intermediary values, leukocytes with very low values and background with high values. Thus, by adding these two color channels, an image was obtained where the erythrocytes and a leukocytes have intermediary values and background with much higher values, making it easier to

differentiate them. The following steps intend to smooth any loss using morphological operations, as described in Algorithm 4. Image 32 shows the complete process.

Algorithm 4 Background removal algorithm:

1: Split preprocessed BGR color space image into Blue, Green and Red channels
2: Sum Blue and Green channels
3: Binarize image using the following rule: IF pixel_value < 255 THEN 0
4: Apply first an Opening then a Closing morphological operation using a kernel (3, 3) with an ellipse shape
5: Remove internal holes with small areas, recovering leukocyte regions taken as background
6: Invert pixel values, making background pixels having 0 value and remaining pixels having 255 value
7: Create a mask changing each pixel value from 255 to 1
8: Multiply the original image with the mask resulting in an image where all background pixels have 0 value

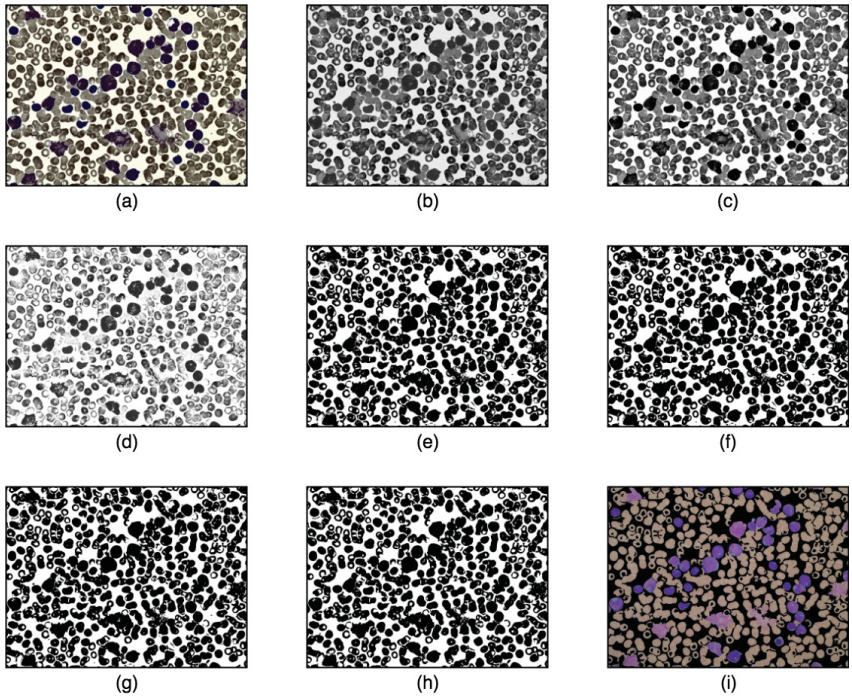


FIGURE 32 – Displays the result of each step of the algorithm 4, (a) preprocessed image, (b) blue channel, (c) green channel, (d) sum result of blue and green channels, (e) binarized image, (f) opening morphological operation, (g) closing morphological operation, (h) internal holes removed, (i) image with background removed. **Source**: the author.

4.5.2 Erythrocyte Removal

Once the background was removed, the only elements remaining in the image were leukocytes and erythrocytes. This step was intended for the removal of all the remaining erythrocytes. When the images with their background removed were analyzed looking for a pattern that distinguishes erythrocytes from leukocytes. The hue channel represents

the structures in a very different way and only leukocytes remain complete, ergo this segmentation pipeline was assembled using this channel.

In some lighting patterns, the hue channel maintained almost only leukocytes, and in cases where erythrocytes were still preserved, they were not complete and could easily be removed by applying an erosion morphological operation, as described in Algorithm 5 and illustrated in Figure 33, thus resulting in an image composed only of leukocytes.

Algorithm 5 Erythrocyte removal algorithm:

1: Convert image with background removed to HSV color representation model
2: Binarise Hue channel using Otsu's method
3: Apply an Erosion three consecutive times using a kernel (5, 5) in the shape of an ellipse
4: Remove all objects with an area smaller than 2000 pixels
5: Create a mask that maintain only leukocytes by changing pixel values from 255 to 1
6: Multiply original image with the mask

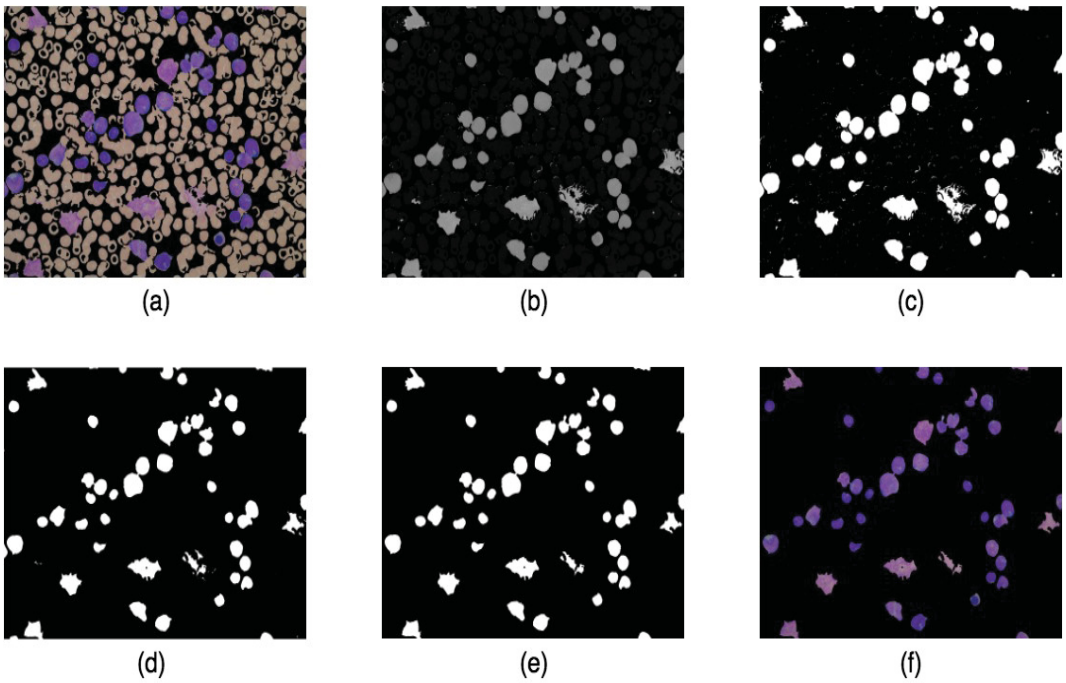


FIGURE 33 – Displays the result of each step of the algorithm 5, (a) background removed image, (b) hue channel, (c) binarized image using Otsu's method, (d) eroded image, (e) small objects removed, (f) multiplication result between the original image and the generated mask. **Source**: the author.

4.5.3 Incomplete Leukocyte Removal

As only complete leukocytes must be taken into account when classifying an image, it was necessary to find a way to remove the ones that were incomplete by being in the extremities of the blood slice when the image was taken.

After the observations, it was noticed that in some images there were complete leukocytes attached to incomplete leukocytes that were on the borders and should be

removed. Therefore, a simple removal of elements that were in contact with the borders removed both attached leukocytes. So this was worked in a way to find where the edge of a leukocyte was, so that we could remove only the incomplete leukocyte and keep the complete one.

The watershed method works very well in this scenario of finding the correct edge for the leukocyte that must be removed. After the edges are computed, it becomes an easy task to remove the incomplete leukocytes by applying a flood fill in each contour that was in contact with the image limits, as described in Algorithm 6.

Algorithm 6 Incomplete leukocyte removal algorithm:

1: Convert image with the background and erythrocytes removed to grayscale
2: Binarize image changing to 255 any pixel with a value higher than 0
3: Verify if there is any object in contact with any extremity
4: **if** there is an object in contact with any extremity **then**
5: Apply watershed on the image with the background and erythrocytes removed
6: Flood fill any contour that makes contact with an extremity
7: Subtract the binary image created in step 2 with flooded image
8: Apply an opening morphological operation with a kernel (3, 3) with an elliptical shape
9: Create a mask that maintains only complete leukocytes by changing pixel values from 255 to 1
10: Multiply original image with the mask
11: **end if**

To illustrate the steps of this pipeline, a different image was chosen than the one used before, because that image doesn't have any attached leukocytes on its extremities. So to better illustrate this algorithm behavior, an image that contains this characteristic was chosen as can be seen in Figure 34.

4.6 SINGLE LEUKOCYTE DETECTION

After obtaining an image composed only of complete leukocytes, the focus changed to extract them individually, since the classification phase was designed to work only with individual leukocytes. As a considerable number of images contains leukocytes attached to them, it was necessary to design a pipeline in order to find the approximate Center Of Mass (COM) of each leukocyte. This section will describe the developed method.

The two concepts defined during the development were *agglomeration* and *attached*, where *agglomeration* refers to a group of leukocytes, formed by more than two leukocytes and *attached* defines the small groups formed by only two leukocytes, as exemplified in Figure 35. Through these concepts, two constants were defined, where an *agglomeration* is an object with an area bigger than three times the area of an isolated leukocyte, and an *attached* is an object with an area bigger than 1.5 times the area of an isolated leukocyte, and smaller than three times the area of an isolated leukocyte.

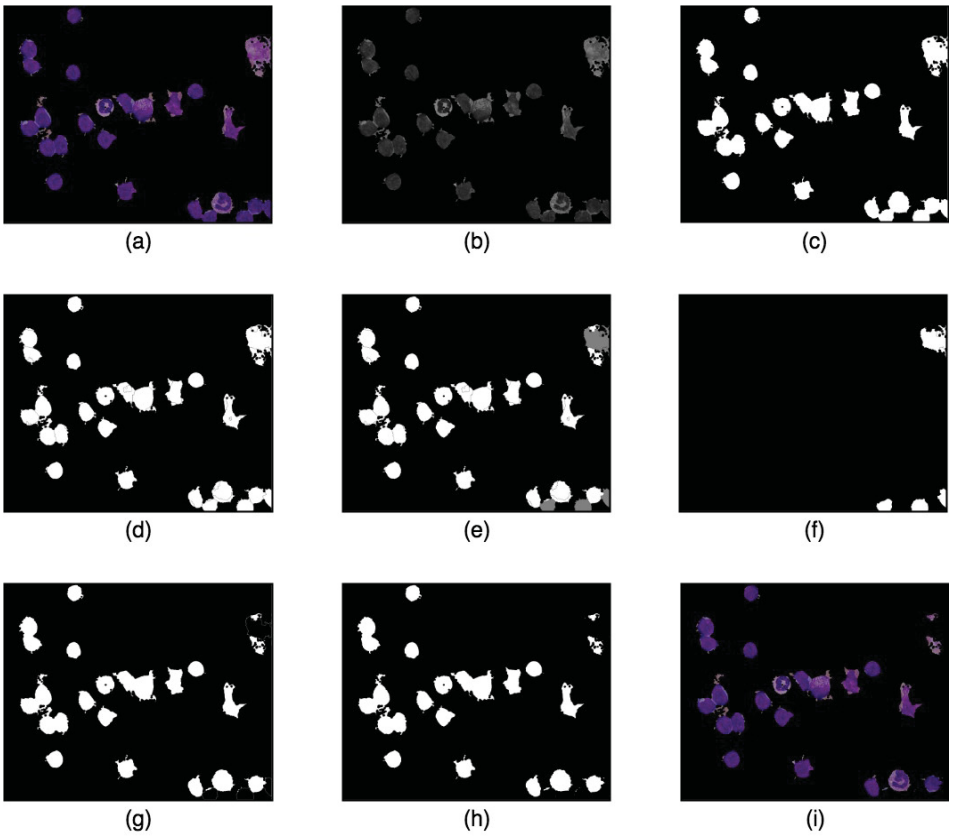


FIGURE 34 – Displays the result of each step of the Algorithm 6, (a) image composed with only leukocytes, (b) grayscale image, (c) binarized image, (d) watershed contours draws in (c), (e) border leukocytes flooded using the 127 value, (f) extracted border leukocytes, (g) subtraction of (c) minus (f), (h) morphological opening of (g) result, (i) image composed only of complete leukocytes. **Source**: the author.

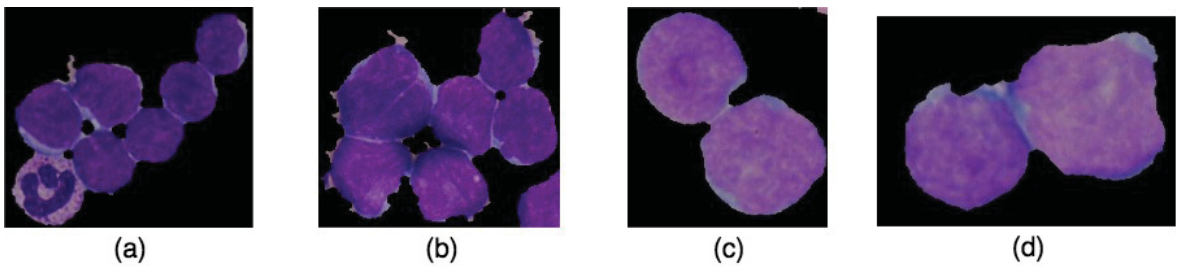


FIGURE 35 – Examples of agglomerations in (a) and (b) and attached leukocytes in (c) and (d). **Source**: the author.

The method designed to individually extract each leukocyte of an image, initially isolates, in separate images, the three possible existing classes, i.e., isolated leukocytes, *agglomerations* and *attached*. Then the COM of each leukocyte was found by dismantling *agglomerations* and *attached*, to finally cut out sub-images.

4.6.1 Discover Median Leukocyte Area

As the ALL_IDB1 database has images in different levels of zoom, using a constant value to represent a single leukocyte area wouldn't perform well in the entire database, and there was no way to find a rational value for this hypothetical value. So it was necessary to find a measure that represented well the size of a single leukocyte in the current image and not for the entire database. The median measure was the one chosen, because the presence of outliers doesn't affect its value as much as in the average, where this outliers were agglomerations or attached leukocytes.

In order to reach the Leukocyte Median Area (LMA) in the images that had *agglomerations* and/or *attached* it was necessary to find and remove them, and calculate the median when there were only isolated leukocytes in the image. Algorithm 7 describes the steps used to calculate the LMA value on the segmented images composed only by leukocytes resulted from the method described in Section 4.5.

Algorithm 7 Median area calculus algorithm:

```
Binarise segmented image
Remove small objects
Fill internal holes in the objects
Calculate median area of all objects
if There any a object with area bigger than 3 times median is founded then
    Find objects with an area bigger than 3 times the median value and remove them
    Recalculate the median area of all objects again
end if
if Any object with an area bigger than 1.5 times the median is founded then
    Find objects with an area bigger than 1.5 times the median value and remove them
    Recalculate the median area of all objects again
end if
```

4.6.2 Detach Leukocytes

This method was applied to several morphological erosion operations and after each erosion, it was verified if one or more leukocytes were detached based on their area, detailed in Algorithm 8. The kernel size used was calculated based on the median leukocyte area, where the kernel size was defined by Equation 4.1, using the median leukocyte area as reference. This way the kernel size adapts to the size of the leukocytes that compose this particular image, and won't suffer from different zoom levels.

$$kernel\_size = \begin{cases} \frac{median\_area}{1000} + 1, & \text{if} \quad \text{mod } (\frac{median\_area}{1000}) = 0 \\ \frac{median\_area}{1000}, & \text{otherwise} \end{cases} \tag{4.1}$$

The area was a criterion adopted, in order to define if an object was an agglomeration or a single leukocyte, and after the application of an erosion all objects become

smaller, therefore this criterion must be updated after every iteration, thus avoiding that after some erosion, an agglomeration is erroneously considered a single leukocyte, Equation 4.2 shows how this value was updated after each erosion.

$$reference\_value = (1 - (1 * kernel\_size * iteration)) * leukocyte\_median\_area \quad (4.2)$$

Algorithm 8 Leukocyte detach algorithm:

```
1: Define kernel size
2: for iteration = 0; iteration < 150; iteration += 1 do
3:     Calculate current reference_value and area of all objects
4:     Select objects with area smaller than reference_value
5:     Compute center of mass of each object selected
6:     Remove selected objects from image that is been processed
7:     Apply erosion operation
8: end for
```

4.6.3 Isolate Different Classes

Once LMA was found, it was possible to assign a class to each object based on its area, where: the classes were *isolated*, *attached* and *agglomerations*. Then each class is isolated to a separated image. The method consists in computing each object area, then verifying which class it belongs to based on Equation 4.3.

$$class = \begin{cases} AGGLOMERATION, & \text{IF } object\_area > (3 * medial\_leukocyte\_area) \\ ATTACHED, & \text{IF } object\_area > (1.5 * medial\_leukocyte\_area) \\ ISOLATED, & \text{OTHERWISE} \end{cases} \quad (4.3)$$

As can be seen in Figure 36, the rule created to distinguish each class based on its size is not perfect, where Figure 36.b has attached leukocytes being defined as agglomerations. It is important that it always *error* to the class above, for example classifying an *attached* as *agglomeration* or an *isolated* as *attached*, because *isolated* leukocytes don't pass through any detach method. It is also is important that methods designed to *detach agglomerations* and *attached* are able to handle this wrong classification problem.

4.6.4 Nucleus Segmentation

As observed in Figure 35, *agglomerations* usually are formed by leukocytes that have their extremities in contact, so when removing leukocyte cytoplasm some leukocytes could be detached, or their connection reduced. It must be highlighted that this method is

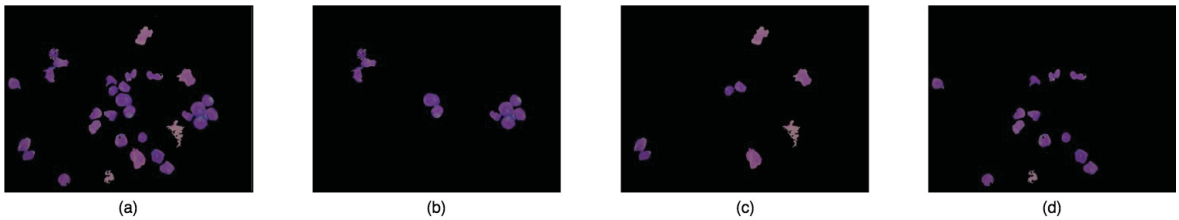


FIGURE 36 – Result of isolation method, (a) segmented leukocyte image, (b) image agglomeration, (c) attached image, (d) isolated leukocytes image. **Source**: the author.

applied only in *agglomerations*, once it is more difficult to find individual COM in them that is *attached*, being unnecessary to apply this method in this case.

This method works with channels of different color representation models, because as they represent an image with different interpretations, combining them could simplify the segmentation process, as detailed in Algorithm 9. The channels selected to be used on highlighted cell nucleus from the cytoplasm are illustrated in Figure 37.

Algorithm 9 Nucleus segmentation algorithm:

1: Convert segmented leukocyte image to HSV model and extract Saturation channel
2: Convert segmented leukocyte image to L*a*b model and extract b channel
3: Convert segmented leukocyte image to YUV model and extract U channel
4: Sum b and U channels
5: Binarize added image using Otsu's method
6: Apply k-means in Saturation channel image, with K = 3 because there are only three classes in the image, i.e., cell nucleus, cytoplasm and background
7: Find the biggest value on k-means resulting image, because Saturation channel tends to represent cell nucleus with values bigger than cytoplasm
8: Binarize k-means result image using the following rule: IF value = biggest_value THEN 0 ELSE 255
9: Apply logic operation AND using images resulting from steps 5 and 8
10: Apply closing morphological operation using a kernel (5, 5) in the shape of ellipse
11: Remove remaining small objects, then invert image, making 255 values represent leukocyte pixels
12: Change 255 values to 1, then multiply this image to original image with complete leukocytes

4.6.5 Find Individual Leukocyte Centers of Mass on *Attached*

After detecting *attached* leukocytes (*attached*), two methods were elaborated, to find the COM of each leukocyte, where the first one was through the application of Algorithm 8 on the nucleus segmented image, and the second one computing the COM of the complete *attached* object.

It was necessary to use different ways to compute COM, due to the fact that the rule defined to distinguish each class isn't perfect, so in some images there were leukocytes with disproportional size in relation to others. So, this method must work taking to account

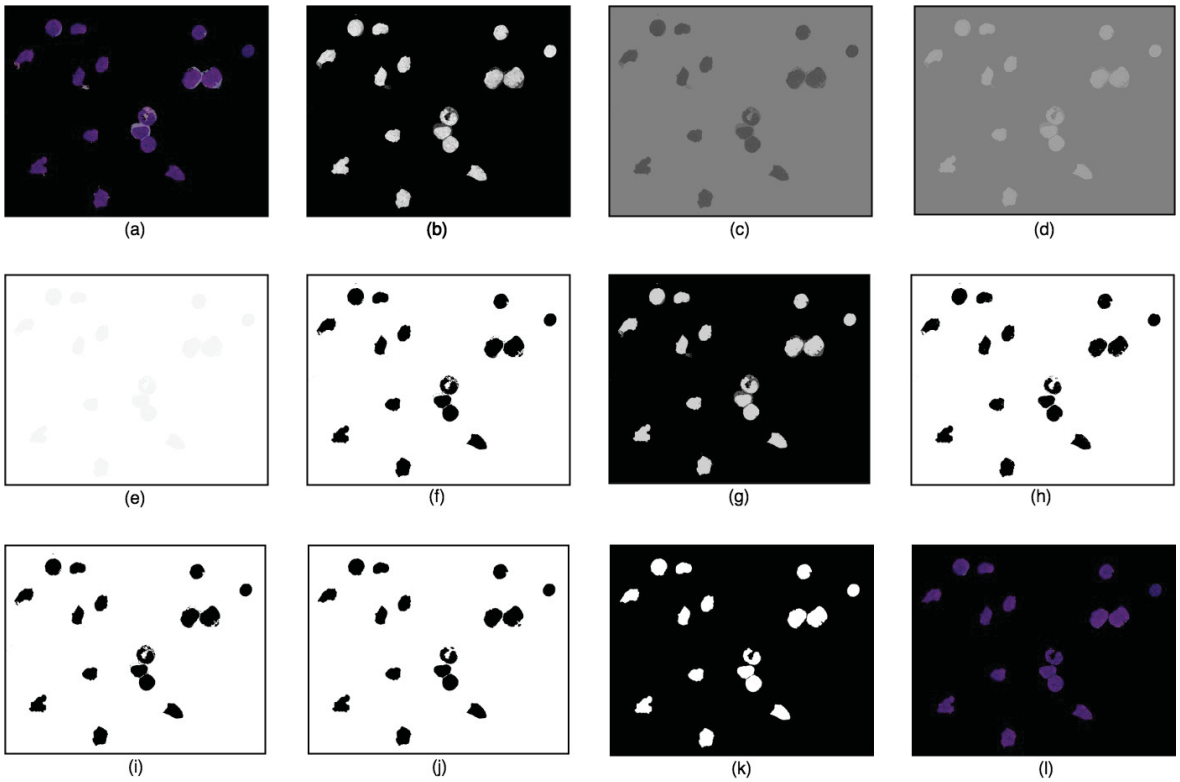


FIGURE 37 – Nucleus segmentation pipeline results, (a) segmented leukocyte image, (b) Saturation channel, (c) b channel, (d) U channel, (e) result of sum of (c) and (d), (f) binarization of (e), (g) k-means applied on Saturation, (h) k-means image binarized, (i) AND logical operation applied on (f) and (h), (j) result of closing operation, (k) inverted image and small objects removed, (l) segmented image. **Source**: the author.

this fact, validating each COM found. This validation consists of removing COM with a *Euclidean Distance* lower than 1.25 times LMA.

Credited levels of confidence for each method were used to extract COM, where the most reliable *attached* cases were the ones found by Algorithm 8, thereby, if two COMs had an *Euclidean Distance* lower than 1.25 times LMA, the one found using erosion in the nucleus was preserved and the other one removed.

This way, one method compensates for the other's failures as seen in image 38, where the red circles in 38(b) represent the COMs using Algorithm 10, the yellow circles in 38 (c) illustrate all COMs found using complete attached leukocytes, and 38 (d) shows the final COMs computed in 38 (a), after the validation makes it clear that the methods compensate each other.

4.6.6 Find Individual Leukocyte Center of Mass in Agglomerations

As *agglomeration* is more complex than *attached*, the same method used to find individual leukocyte centers of mass in *attached* was incremented to fit this class, where the main difference is that there is one more way to compute COM, as described in Algorithm

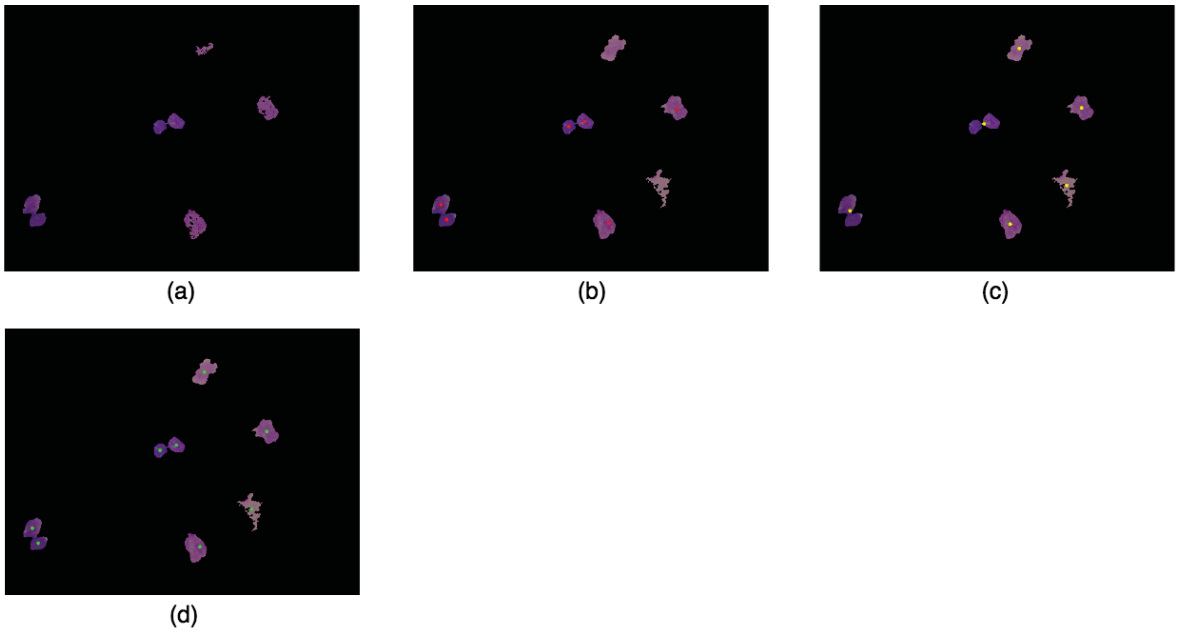


FIGURE 38 – Nucleus segmentation pipeline results, (a) Leukocytes defined as *attached*, (b) COM found using Algorithm 8 marked in red, (c) COM using complete object, marked in yellow, (d) COM found after combining and validating COM of (b) and (c), marked in green. **Source**: the author.

10. This method consists of applying Algorithm 8 to the agglomeration image without segmenting its nucleus, thus generating a different COM, then following the same steps as the method used in *attached*.

The trust rating of the methods used to calculate COM changed in relation to the one applied in *attached*, where COM found through the original agglomeration image was given the highest credibility, then COM was found in the nucleus segmented image, and finally, COM of the complete objects was also found. As observed in Figure 39, the three methods complement each other.

Algorithm 10 Nucleus segmentation algorithm:

1: Isolate agglomerations in one image
2: Apply algorithm 8 in the agglomeration image and store COM found
3: Segment nucleus present in the agglomeration image
4: Apply algorithm 8 in the nucleus segmented image and store COM found
5: Remove closed COM, prioritizing the ones found in step 2
6: Compute COM of the complete objects
7: Remove closed COM, prioritizing the ones found until step 5

4.6.7 Sub-image Extraction

As the goal was to replicate the ALL_IDB2 image pattern, where the images have one central leukocyte that was considered the object of interest, as detailed in Section 4.1.2, in the ALL_IDB1 images, leukocytes were classified individually for a classifier

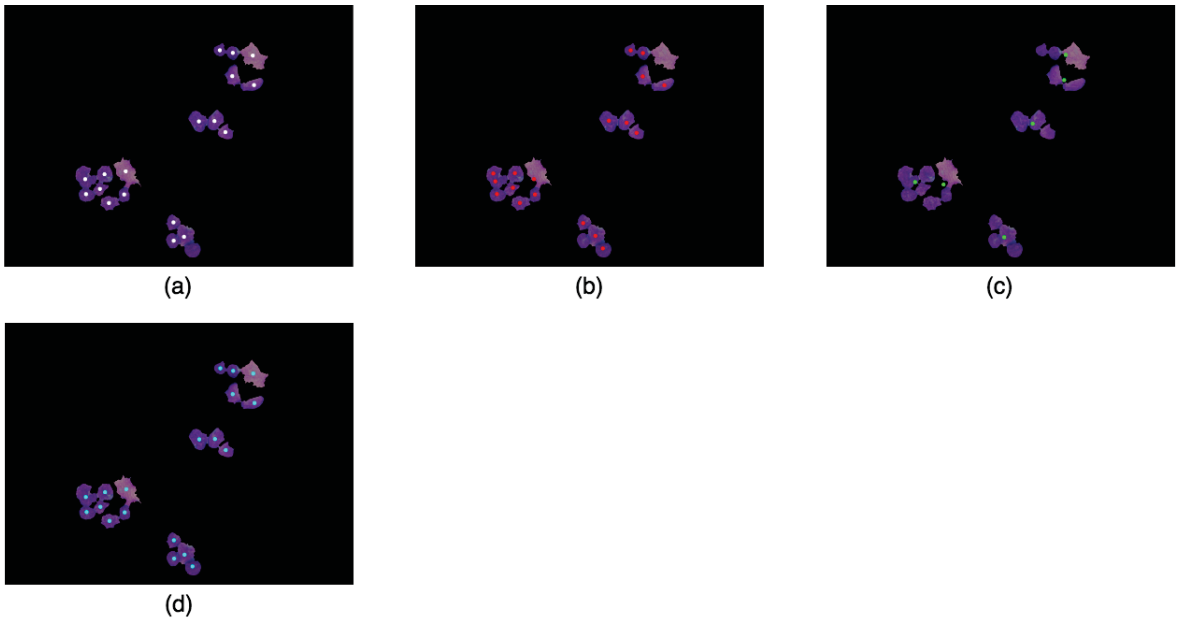


FIGURE 39 – Nucleus segmentation pipeline results, (a) COM marked in white by Algorithm 8 applied in the complete agglomeration image, (b) COM marked in red found using Algorithm 8 in the nucleus image, (c) COM using complete object, marked in green, (d) COM found after combining and validating COM of (a), (b) and (c), marked in blue. **Source**: the author.

trained with the ALL_IDB2 images. It must be highlighted that the sub-images were extracted from the original ALL_IDB1 image without any segmentation.

After estimating the COM of each leukocyte in an ALL_IDB1 image, the sub-images were cut out, where the central point of a sub-image corresponds to a COM of a leukocyte. As the ALL_IDB2 database doesn't have a single image size, it was decided to use the most common size which is 257 x 257. In cases where the leukocyte was too close from an edge, the central point of the sub-image was dislocated until it was possible to reach a 257 x 257 size, even though the image central point wasn't the same as the leukocyte central point, which also happens in the ALL_IDB2 database, so it doesn't compromise the classification as the classifier deals with this case in the training process.

4.7 SUB-IMAGE AND ALL_IDB2 SEGMENTATION

As one type of feature used in the classification was texture, and in this particular kind of feature it performs better if it was acquired from an image containing only the object of interest, a method was developed in order to segment only the central leukocyte present in an image. This method was applied in sub-images extracted from the ALL_IDB1 images and the ALL_IDB2 database.

A large part of the methods used in ALL_IDB1 leukocyte segmentation were reused since both sub-images and ALL_IDB2 images come from the ALL_IDB1 images, as pointed in the Algorithm 11. The difference between this method and the one used

to segment ALL_IDB1 images, described in Section 4.5, is that in this case only the leukocyte that was closer to the center must remain in the image, so an additional step to remove any leukocyte which was not the central was added.

Algorithm 11 Central leukocyte segmentation algorithm:

1: Remove background, using method described in Section 4.5.1.
2: Remove erythrocytes, using method described in Section 4.5.2.
3: Remove incomplete leukocytes in contact with edges, using method described in Section 4.5.3
4: Detect and drawn on black leukocyte edges, using watershed algorithm. This is used to deal with attached leukocytes.
5: Detach leukocytes using an erosion morphological operation, with kernel (5, 5) in ellipse shape
6: Compute center of mass of each detached leukocyte.
7: Calculate Euclidean distance between central point of the image and center of mass of each leukocyte
8: Remove all leukocytes, except the one with the shortest distance to the center.

In more complex cases some part of the central leukocyte could be lost, as can be seen in Figure 40, this usually happens in sub-images where the central leukocyte is part of an *agglomeration* or an *attached*.

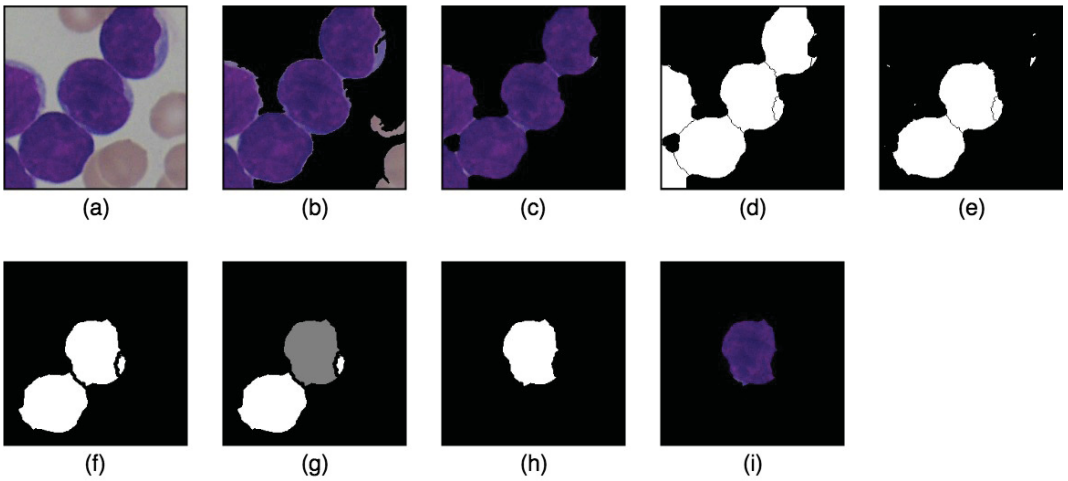


FIGURE 40 – Sub-image segmentation method results, (a) sub-image extracted from ALL_IDB1 image, (b) background removed, (c) erythrocytes removed, (d) image (b) binarized with edges found by watershed drawn, (e) incomplete leukocytes removed by flooding extremities, (f) erosion applied on (e) to detach objects, (g) central leukocyte found, highlighted in gray, (h) all leukocytes removed except the central, (i) segmented image acquired by using (h) as a mask and multiply it by (a). **Source**: the author.

4.8 FEATURE EXTRACTION AND SELECTION

Once the detection extraction of sub-images and segmentation was done, which features would be used to distinguish leukocytes with ALL from healthy ones started to

be analyzed. After analyzing research papers that worked with the same objective, it was noticed that features extracted using CNN, texture and morphological features presented the most relevant results. However none of the papers combined both feature extraction techniques.

4.8.1 Texture Features

Texture features are used in several computer vision applications with many different purposes, and in most cases they perform very well. In the leukocyte classification case, this is not an exception either. Different from other papers that convert the image to gray scale instead of extract texture features, in this case features were computed from any of the *YUV*, *HSV*, *BGR* and *L*a*b* channels representation models, using the *GLCM* (HARALICK et al., 1973) and *LBP* (OJALA et al., 1996) methods on sub-images after segmenting the central leukocyte.

4.8.2 CNN Features

Transfer learning is the area in machine learning which studies how to apply a knowledge learned in a specific scenario to a different one. A successful application of this hypothesis is seen by using a Convolutional Neural Network (CNN) trained with a database A to extract features from a database B, and the results are very consistent, as observed by (RAZAVIAN et al., 2014).

The ResNet50 CNN architecture introduced by (HE et al., 2015) was used as a feature extraction method which generates feature vectors with 2048 features. It must be pointed out that the CNN was trained using the *ImageNet* database (DENG et al., 2009) before extracting the features, and the features were extracted from the sub-images without segmenting them.

4.8.3 Morphological Features

After analyzing the five classes of leukocytes, it was noticed that their morphology was very different, and most of the articles found in the literature used morphological features and presented very consistent results, as seen in the work proposed by (PIURI; SCOTTI, 2004), that uses only morphological features to classify leukocytes according to their lineage. Only three morphological features were computed, i.e., *Area*, *Perimeter* and *Ratio between the area of the nucleus to the area of the cell*.

4.8.4 Feature Vector

Once the feature extraction phase ended, a total of 5279 features were extracted, so it was necessary to reduce them by selecting the most relevant ones, thus avoiding the course of dimensionality, since there were only 260 samples used to train the classifier.

Before applying the dimensionality reduction, the feature vectors were normalized using the *Z-score* (LARSEN; MARX, 2000) method, so none of the features received a bigger weight only because they had a bigger magnitude of values.

The feature selection phase uses the Recursive Feature Elimination (RFE) (GUYON et al., 2002) technique to find the n most important features, thus being able to select a smaller subset formed by the n most relevant features to represent each sample. RFE uses a linear regression model as a Support Vector Regression (SVR) to estimate the importance of each feature and remove the least important at each iteration, until there are only n features remaining.

Different sizes of feature vectors, selecting a group with the n more relevant features found by RFE. The tests started using only two features and grew their size by using more features until a consistent classification result was found. The smallest feature vector that present the highest accuracy rate was made of only 15 features illustrated in Figure 41, where seven of them were extracted using texture, and eight using CNN. An interesting fact is that none of the morphological features were used.

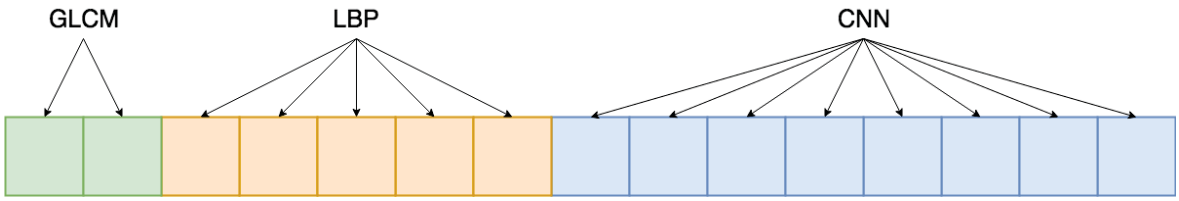


FIGURE 41 – Shows how many features of each method was used for the final feature vector. **Source**: the author.

4.9 IMAGE CLASSIFICATION

With the definition of the definitive feature vector, the classification phase started. As observed in other works that used the same database, the authors used a global approach by extracting a feature vector related to the complete image, as seen in (VOGADO et al., 2017a). Our approach focused on classifying an image based on the features extracted from the regions of interest, which in this case were the leukocytes of an image.

Since cancer cells tend to reproduce themselves in a disorderly fashion, a premise was made that *an image from a healthy patient has more leukocytes classified as negative than leukocytes classified as positive to ALL* (PERINI, 2016). The premise used to define after the individual classification of the leukocytes was made not so restrictive on purpose, because a single leukocyte wrongly classified doesn't compromise the classification of the entire image.

As described in Section 4.2, the classifier was trained using the feature vectors extracted from the all ALL_IDB2 images that were not sub-images from the current ALL_IDB1 image. Different kinds of classifiers were tested in order to find which one

generates the best results. Classifiers were based on optimization, such as *Support Vector Machines* (CORTES; VAPNIK, 1995), *Perceptron* (ROSENBLATT, 1958) and *Multilayer Perceptron* (HAYKIN, 2001), distance based *K-nearest neighbor* (AHA et al., 1991), search (FISHER, 1938), *Decision Tree* (L FRIEDMAN J, 1984) and probabilistic *Linear Discriminant Analysis*.

All classifiers used had parameters that impacted on the classifier performance. The best parameters of each classifier were found using exhaustive search implemented on the sklearn class *sklearn.model_selection.GridSearchCV* (PEDREGOSA et al., 2011). Table 3 displays the parameters found for each classifier.

TABLE 3 – Parameter table for classifiers.

Classifier	Parameters
SVM	C = 0.1, kernel = linear
KNN	k = 5, algorithm = 'ball_tree'
LDA	solver = 'Least squares solution'
Decision Tree	criterion = 'entropy', mass_depth = '20',
Perceptron	penalty = 'l2', max_iter = 2000
MLP	activation = 'relu', hidden_layers = (5,), solver = 'lbfgs', momentum = '0.9'

5 RESULTS AND DISCUSSION

In this chapter, the results for each major phase proposed in this work are presented. Also each result from the tests performed are discussed and compared with literature articles containing similar proposals.

5.1 SEGMENTATION RESULTS

As some features were extracted from images assuming that they contain only leukocytes, it was crucial that this premise was largely true, so as not to compromise the classification phase. This section presents the results of segmentation methods described in Section 4.5 and Section 4.6, thus allowing an analysis of its performance.

5.1.1 ALL_IDB1 Leukocyte Segmentation

The method described in section 4.5 has succeeded in correctly segmenting only complete leukocytes in 92.6% of the complete ALL_IDB1 database. From the images with any segmentation error (7.4%) only 2.8% suffer with a leukocyte loss. In these cases not all leukocyte were lost in one image, making it possible to classify this image based on the remaining leukocytes.

The biggest concern was to build a methodology that didn't have a any bias, thus avoiding the chance of it performing well only on images used during its development. In order to verify that this bias doesn't exist, the ALL_IDB1 database was divided in two image groups as mentioned in section 4.3, and when applied on the image subset used to validate it, the results were consistent, not varying much between one database and another, as illustrated in Figure 42.

After analyzing the segmentation results, it was concluded that the method reached its objective, performing well under different lighting patterns, almost with the same results as illustrated in Figure 43, different from other works that use the same database, where they usually use only a small subset of images, because they have the same lighting pattern, like the method proposed by (PUTZU et al., 2014), which uses only 33 images of the base.

5.1.2 Individual Leukocyte Detection

The correct individual leukocyte detection was an important phase, since the classification of a complete image is based on its individual leukocyte classification, so any leukocyte lost would have a huge impact.

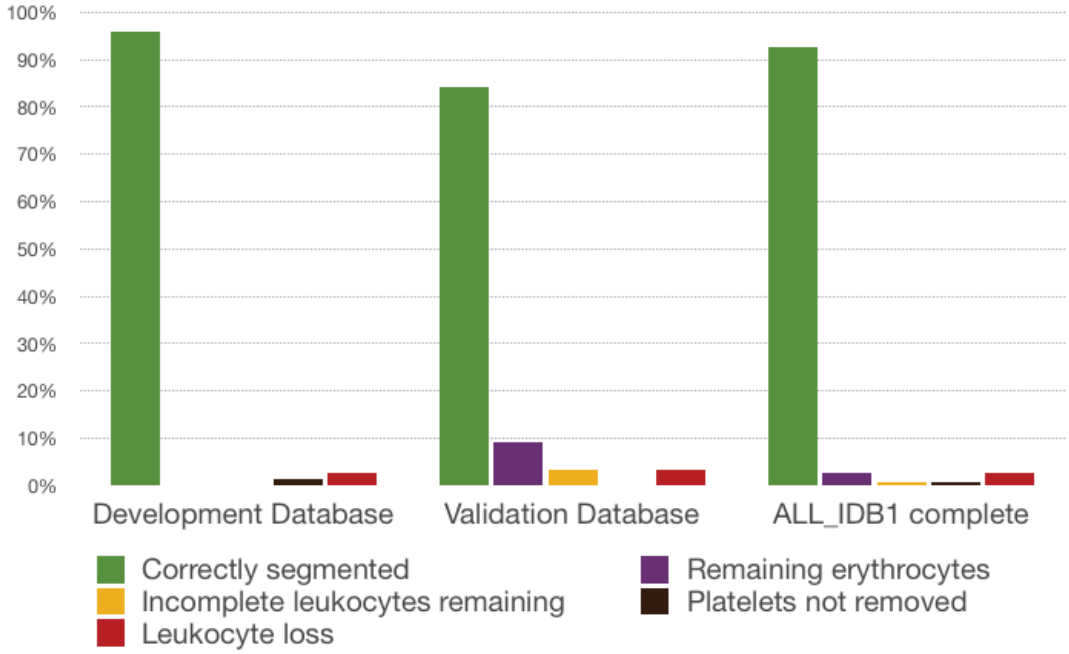


FIGURE 42 – Displays segmentation results of the method described in section 4.5 on each subset and the complete ALL_IDB1 database. **Source**: the author.

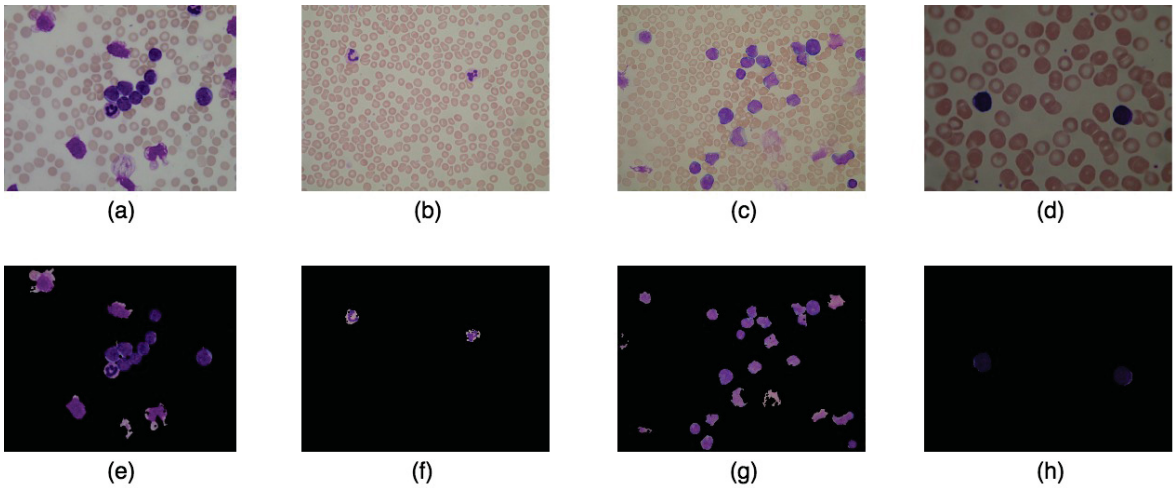


FIGURE 43 – Exemplifies segmentation pipeline results in different lighting patterns, where the images on the second line (e, f, g and h) are the segmentation result of above images (a, b, c and d). **Source**: the author.

As detailed in Section 4.6, the biggest adversity found was to estimate the individual COM (Center Of Mass) of leukocytes that compose *agglomerations* and *attached*. However, the results of this method showed to be very consistent, once 78.7% of the complete ALL_IDB1 database had all its leukocyte COM detected correctly, as displayed in Figure 44.

Two kinds of errors were found in this process. One happens when a single leukocyte has more than one COM attributed to it. However, this error has lower impact than the other one that happens when an *agglomeration* or an *attached* doesn't detached

correctly, attributing a single COM to two or more attached leukocytes, and this error occurs in only 2.8% of the images in the complete ALL_IDB1.

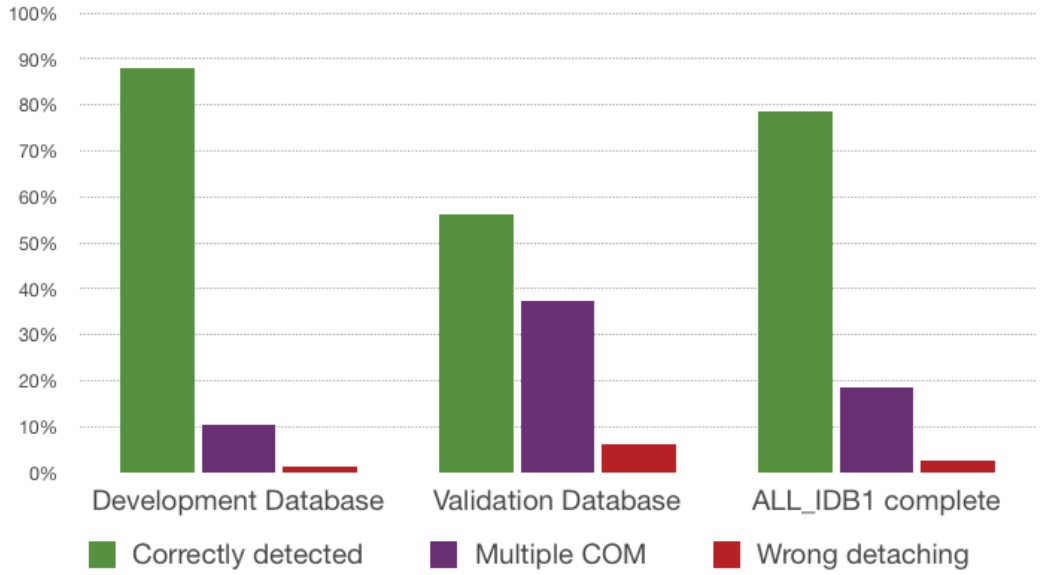


FIGURE 44 – Displays single leukocyte detection results from the method described in section 4.6 on each subset and the complete ALL_IDB1 database. **Source**: the author.

The multiple COM error occurred in two scenarios, illustrated in Figure 45, where the most common error occurred when an object were defined as *agglomerations*, *attached* or *isolated*, and the least common occurs when in some images there were leukocytes of the following categories: *neutrophil*, *eosinophil* and *basophil*. Due to the fact that its nucleus does not have a circular shape like *lymphocyte*, it should be strengthened that ALL manifests only on *lymphocytes*.

5.1.3 Sub-images and ALL_IDB2 Segmentation Results

As the features extracted from an image depend on the image provided, this phase must perform well so the classification phase won't be compromised. Images with attached leukocytes were repeatedly the ones that present more difficulties in segment the central leukocyte with minimal information loss, and presenting almost 90% of the ALL_IDB2 database and the sub-images extracted from the ALL_IDB1 images segmented correctly. This phase could be considered a success, as illustrated in Figure 46.

In almost only 5% of the images, the central leukocyte was completely lost, but in none of the ALL_IDB1 images, all leukocytes were lost, for example in the image Im005_1 from the ALL_IDB1 database, which was the image with more leukocytes lost, 31 leukocytes were detected, from which 5 were lost, 1 presented some information loss and 25 were correctly segmented, so it can be concluded that the classification of this image was not compromised.

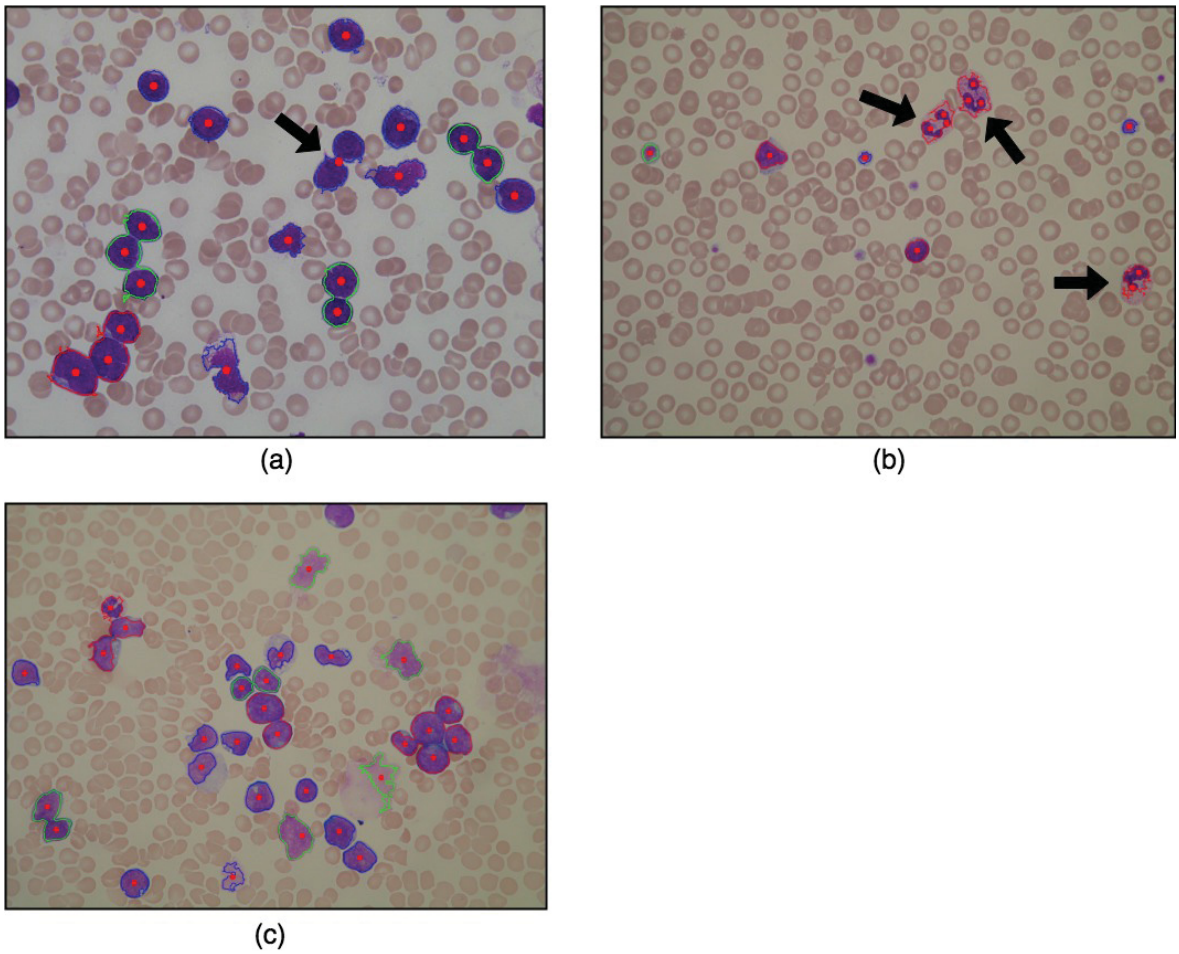


FIGURE 45 – Illustrates leukocyte individual COM detection results, blue contours represent *isolated*, green refers to *attached* and red contours identify *agglomerations*. (a) contains an error due to wrong attribution of what should be *attached* as *isolated* (black arrow). (b) image containing *neutrophil* which in this case has multiple COM, detected a loss of a single leukocyte (black arrow). (c) image with *agglomerations*, *attached* and *isolated* attributed correctly, and all leukocyte COMs computed correctly even with a *neutrophil* being present in this image. **Source**: the author.

As the definitive classifier used was *SVM*, theoretically, a small number of cases that are very different from other examples of the same class doesn't impact that much on the decision boundary, once it attributes greater weight to samples that are closer to the border that faces other class borders (support vectors). Thereby, the classification phase could deal with a small number of leukocytes wrongly segmented without generating a huge impact.

5.2 DIMENSIONALITY REDUCTION

As described in Section 4.8.4, different sizes of feature vectors were tested before defining their final size. The tests consisted in using an *SVM* classifier to classify ALL_IDB1 images using different feature vectors and stop increasing its sizes when the accuracy

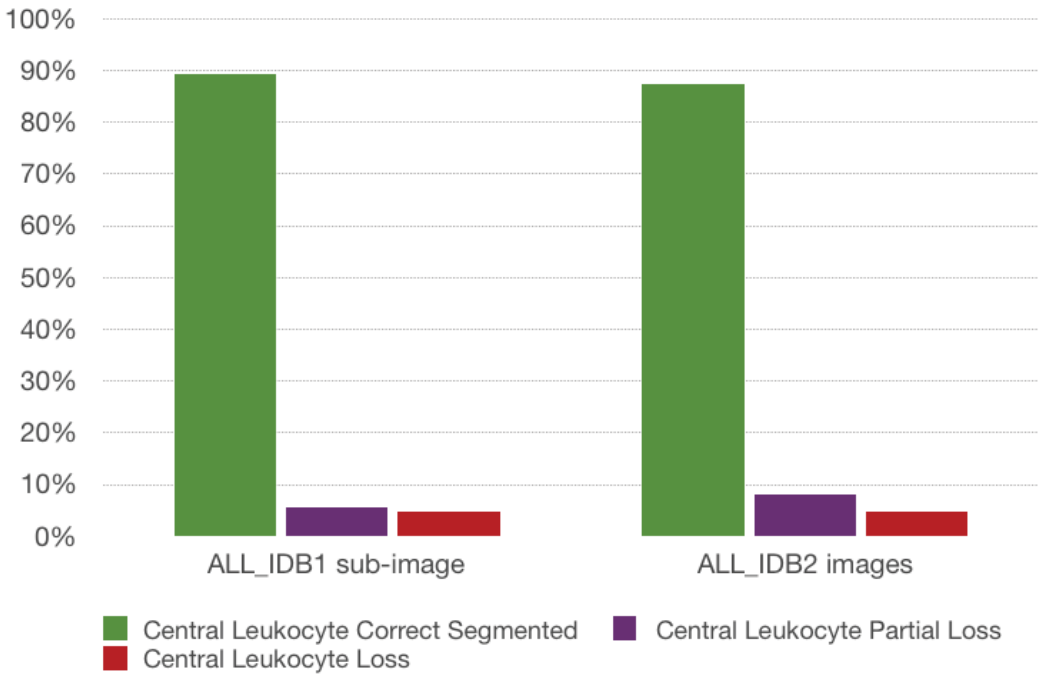


FIGURE 46 – Central leukocyte segmentation results of method described in section 4.7 on ALL_IDB2 complete database and ALL_IDB1 sub-images extracted. **Source**: the author.

stabilized. As illustrated in Figure 47, the accuracy stabilized using 15 features, which is less than 1% of the features extracted.

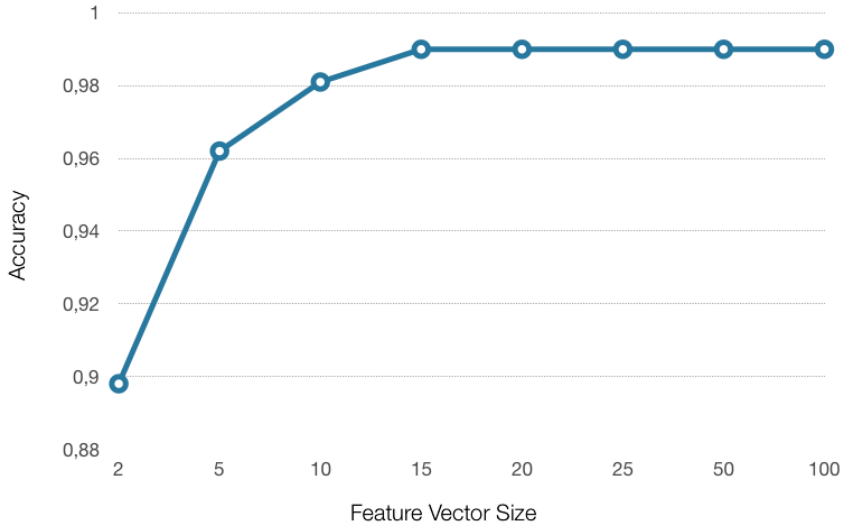


FIGURE 47 – Illustrates the test results done to visualize the impact of the feature vector size. **Source**: the author.

As the results presented by the classifiers were very high on the development base as seen in Figure 48, with *SVM* hitting the entire development database. Thereby, it wasn't necessary to combine classifiers looking for better results. Instead, the classifier with the best accuracy was chosen and used alone.

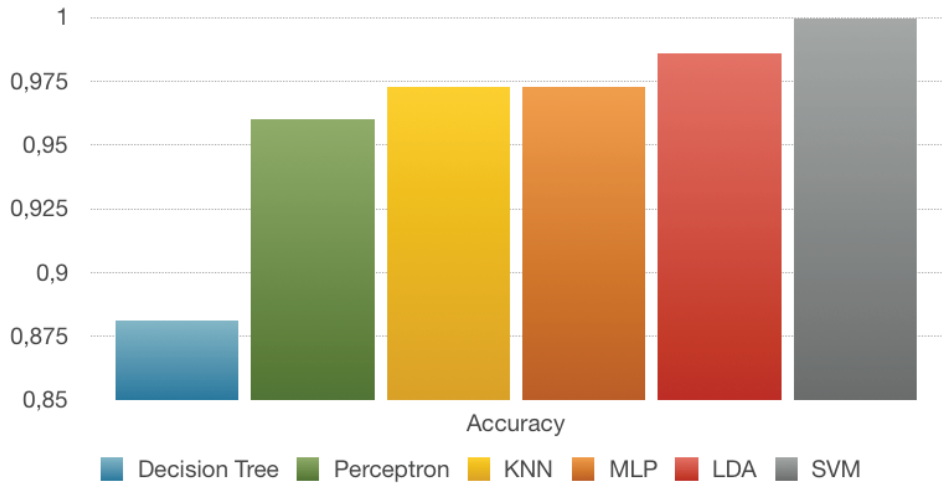


FIGURE 48 – Shows the accuracy of each classifier tested on development database. **Source**: the author.

5.3 ISOLATED FEATURES

Theoretically, features extracted using different methods can generate different representations for the same object, combining features extracted using distinct methods, then selecting the most relevant one in order to build a feature vector that could perform better than a feature vector made of only one kind of feature.

Another scenario that must be verified is the real necessity of using distinct methods of feature extraction, verifying that using only one kind of feature, the classes can be completely discriminated from each other.

In order to prove these hypothesis, feature vectors made of only one kind of feature (texture, CNN or morphology) were tested, then, an *SVM* classifier was used in order to classify all images of the development database, using feature vectors with the 15 most relevant features of texture, CNN or 3 morphological features extracted. The results illustrated in Figure 49 proved that the methodology of combining different kinds of features, and then selecting the most relevant one, generated an increase on the classifier's performance.

5.4 ALL_IDB1 CLASSIFICATION

Reaching an accuracy of 0.99 in the complete ALL_IDB1 database by *Support Vector Machines Classifier*, the classification phase can be considered successful. Figure 50 displays the accuracy reached by *SVM* in each database, allowing the conclusion that the high score on the development database wasn't the result of overfitting as the validation database presented a compatible result.

The only case of wrong classification was a *false positive*, which wasn't the worse

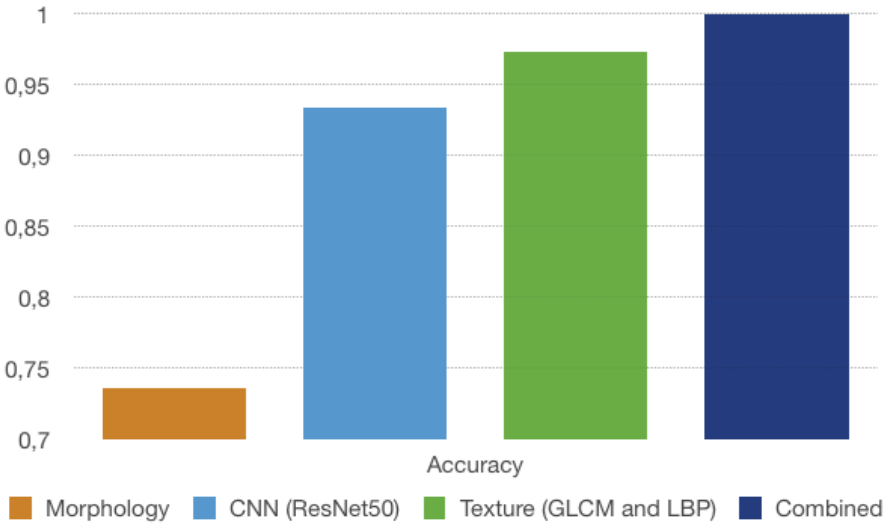


FIGURE 49 – Results of the tests done to visualize the impact combine features extracted using different methods. **Source**: the author.

FIGURE 50 – Results of the *SVM* classifier in each database used. **Source**: the author.

case scenario in the medical applications context, as in a practical scenario, when a false positive happens, the doctor will check more deeply into the case and find that, it was in fact, a false positive.

Observing Figure 51 that displays the segmentation of the image that was wrongly classified, it can be noticed that the problem was in the segmentation phase, as seen in Figure 51 (b), where all the cytoplasm was lost. With the loss of all cytoplasm, the true nucleus was considered as a complete cell compromising the feature extraction phase, so it can be concluded that the segmentation failure triggered the error.

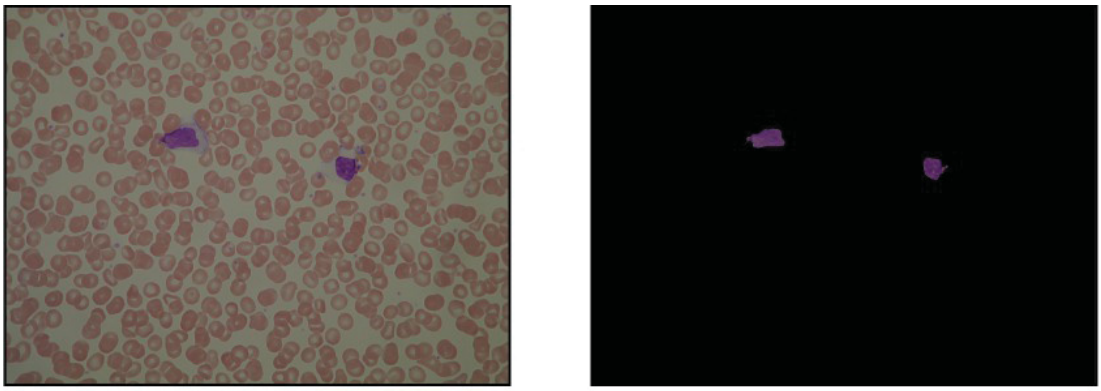


FIGURE 51 – Segmentation result of the wrongly classified image. (a) original image Im107_0, (b) segmented image. **Source**: the author.

6 CONCLUSION

Based on the accuracy reached, it can be concluded that the proposed pipeline accomplished its objective. Even when it generated a wrong classification, this was not the worse case scenario, as it was a false positive which in this context is a lesser problem than a false negative. The accuracy was the only evaluation metrics used, as other related works also used this same evaluation metrics.

For not having access to the masks with leukocyte markings done by specialists, it was not possible to compute *dice coefficient*, *standard deviation*, *distance* or any other measures that evaluate the segmentation results. So this phase was analyzed by observations only, since it was not allowed to make a comparison to the works focused only on the segmentation. Thus it presented these results.

So the comparisons were made with the works focused on *acute lymphoid leukemia* classification that used the ALL_IDB database and, four papers were used to compare the results as they used ALL_IDB1 images in their tests. Table 4 shows the results reached by each paper and the number of images used in the tests, which is a crucial information as one of the characteristics of ALL_IDB is to present distinct lighting patterns and zoom levels, making the development of a single method that works with same accuracy in the complete database a complete different task than one which uses only a subset containing a single lighting pattern or zoom level.

TABLE 4 – Accuracy of classification proposals compared to proposed method. **Source**: the author

Authors	Images used	Accuracy
[Monica Madhukar, 2012]	98	0.935
[Fatma and Sharma, 2014]	50	0.91
[Putzu et al., 2014]	33	0.93
[Vogado et al., 2017]	108	0.981
Proposed method	108	0.99

(VOGADO et al., 2017b) was the only work found that used the complete ALL_IDB1 in its tests, making it the most suitable to make a comparison. There are two big differences between their approach and the one presented on this work. The first one is that (VOGADO et al., 2017b) extracted features from the complete image to make the classification, while our approach aims to isolate and classify each leukocyte individually, and then classify the complete image. The other difference that's while they used only features extracted by *CNNs* our approach uses *CNN features, morphological* and *texture* to compose the feature vector.

Therefore, it can be concluded that a feature vector that combines diverse representation measures, such as *texture*, *morphological* and *CNN features*, generates the most precise representation, being able to use a single classifier and reaching a very high accuracy. Also, the feature selection phase shows that *texture* and *CNN* are more suitable for this particular problem than *morphological* features, which was a common hypothesis, since the specialists classify leukocytes based mostly on their nucleus shape, so the *morphological* features are probably the most suitable to be used in this problem.

6.1 FUTURE WORKS

This section presents the future works that can be done as a sequel of this proposal, making it more comprehensive as it was focused on the ALL detection, but leukocyte and erythrocytes count would be a great addition as a segmentation refinement.

6.1.0.1 Test the proposed method in other databases

During the development of this work only ALL_IDB wsa used, even if presenting several patterns of images, testing the proposed pipeline on images that don't belong to ALL_IDB will generate a better idea if the method works well only on ALL_IDB or if it can be used on other blood smear images without suffering a huge impact.

6.1.0.2 Precise segmentation evaluation

As mentioned before we haven't has access to the markings made by the specialists, so as to precisely evaluate the segmentation phase, the idea is to contact the specialist to do the masks in order to precisely evaluate this phase as well as one of the feature works, once now it is presumed that the segmentation phase works better based on the observations and the classification results.

6.1.0.3 Apply machine learning techniques in segmentation phase

As seen in (LODDO et al., 2016) *machine learning* methods can be used to generate a very precise segmentation, but in their work there was the need of sub images extracted manually in order to extract and train the classifier, which is a semi-automatic method. Based on that, the segmentation phase for this work can be used to automatically segment a great portion of the pixels that belong to each class, to further extract features from them and use a classifier to generate a more precise segmentation.

6.1.0.4 Adapt the method to also count leukocytes

During the bibliography review, some interesting papers were found, which focused on leukocyte counting, which is a more comprehensive approach, due to the fact that there

are many other blood diseases that change the leukocyte population.

REFERENCES

AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. *Machine Learning*, v. 6, n. 1, p. 37–66, Jan 1991. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00153759>. Citado 2 vezes nas páginas 36 e 78.

ALVES, D. R. *Avaliação dos Modelos de Cores RGB e HSV na segmentação de Curvas de Nível em Cartas Topográficas Coloridas*. Dissertação (Mestrado) — Pós-Graduação em Engenharia Elétrica - Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte - MG, Maio 2010. Citado na página 26.

AZEVEDO-MARQUES, P. M. d. Diagnóstico auxiliado por computador na radiologia. *Radiologia Brasileira*, v. 34, n. 16, p. 285 – 293, 10 2001. Citado na página 18.

BARELLI, F. *Introdução a Visão Computacional*. [S.l.]: Casa do Código, 2018. Citado na página 29.

BARRETO, J. et al. Antineoplastic agents and the associated myelosuppressive effects: A review. *Journal of pharmacy practice*, v. 27, 08 2014. Citado 2 vezes nas páginas 7 e 22.

BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 7, p. 711–720, Jul 1997. ISSN 0162-8828. Citado na página 37.

BERGEN, T. et al. Segmentation of leukocytes and erythrocytes in blood smear images. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, v. 2008, p. 3075–8, 02 2008. Citado na página 43.

BEUCHER, S.; MEYER, F. The morphological approach to segmentation: The watershed transformation. *Mathematical Morphology in Image Processing*, Vol. 34, p. 433–481, 01 1993. Citado na página 30.

BRADSKI, G. Open source computer vision library. *Dr. Dobb's Journal of Software Tools*, 2000. Citado na página 40.

CHOLLET, F. et al. *Keras*. 2015. <https://keras.io>. Citado na página 41.

COELHO, L. P. Mahotas: Open source software for scriptable computer vision. *Journal of Open Research Software*, v. 1, July 2013. Citado na página 40.

CORTES, C.; VAPNIK, V. Support vector networks. *Machine Learning*, v. 20, p. 273–297, 01 1995. Citado 2 vezes nas páginas 39 e 78.

DANYALI, H.; HELFROUSH, M. S.; MOSHAVASH, Z. Robust leukocyte segmentation in blood microscopic images based on intuitionistic fuzzy divergence. In: *2015 22nd Iranian Conference on Biomedical Engineering (ICBME)*. Tehran - Iran: [s.n.], 2015. p. 275–280. Citado na página 46.

DENG, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009. Citado na página 76.

DESHPANDE, A. *The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3)*. 2016. <https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html>. Acessado em 09/06/2018. Citado 2 vezes nas páginas 8 e 34.

DJALDETTI, M. et al. Sem observations on the mechanism of platelet release from megakaryocytes. *Thrombosis and haemostasis*, v. 42, n. 2, p. 611—620, August 1979. ISSN 0340-6245. Disponível em: <http://europepmc.org/abstract/MED/505368>. Citado na página 21.

FACELI, K. et al. *Inteligência Artificial, Uma abordagem de Aprendizado de Máquina*. [S.l.]: LTC, 2011. Citado 5 vezes nas páginas 34, 36, 37, 39 e 40.

FARAG, A. Morphological classification of blood leucocytes by microscope images. In: *2003 46th Midwest Symposium on Circuits and Systems*. Cairo - Egypt: [s.n.], 2003. p. 701–703 Vol. 2. Citado 2 vezes nas páginas 51 e 52.

FARIAS, M. G.; CASTRO, S. M. d. Diagnórstico laboratorial das leucemias linfóides agudas. *Jornal Brasileiro de Patologia e Medicina Laboratorial*, v. 40, p. 91 – 98, 04 2004. Citado na página 18.

FATMA, M.; SHARMA, J. Identification and classification of acute leukemia using neural network. In: *2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*. Greater Noida - India: [s.n.], 2014. p. 142 –145. Citado na página 54.

FILHO, O. M.; NETO, H. V. *Processamento Digital de Imagens*. [S.l.]: Brasport, 1999. Citado 4 vezes nas páginas 7, 24, 27 e 30.

FISHER, R. The statistical utilization of multiple measurements. *Annals of Eugenics*, v. 7, n. 2, p. 376–386, January 1938. Citado 2 vezes nas páginas 37 e 78.

GAO, W.; TANG, Y.; LI, X. Segmentation of microscopic images for counting leukocytes. In: *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*. Shanghai - China: [s.n.], 2008. p. 2609–2612. Citado na página 43.

GATH, I.; GEVA, A. B. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 11, n. 7, p. 773–780, Jul 1989. ISSN 0162-8828. Citado na página 30.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. [S.l.]: Pearson, 2010. Citado 5 vezes nas páginas 8, 24, 25, 29 e 31.

GORODNICHY, D.; GRANGER, E.; RADTKE, P. Survey of academic research and prototypes for face recognition in video. 09 2014. Citado 2 vezes nas páginas 8 e 33.

GUYON, I. et al. Gene selection for cancer classification using support vector machines. *Machine Learning*, v. 46, n. 1, p. 389–422, Jan 2002. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1012487302797>. Citado 2 vezes nas páginas 35 e 77.

HAMERSCHIAK, N. *Câncer Infantil - Leucemia Linfoide Aguda (LLA)*. 2016. <http://www.abrale.org.br/leucemia-infantil/lla-infantil>. Acessado em 18/06/2018. Citado na página 23.

HAMERSCHIAK, N. *O que é Leucemia?* 2016. <https://www.abrale.org.br/doencas/leucemia>. Acessado em 17/06/2018. Citado na página 22.

HAO, L. W.; HONG, W. X.; HU, C. L. A novel auto-segmentation scheme for colored leukocyte images. In: *2010 First International Conference on Pervasive Computing, Signal Processing and Applications*. Harbin - China: [s.n.], 2010. p. 916–919. Citado na página 47.

HAO, L. W. et al. A leukocyte nucleus segmentation scheme based on fingerprint smoothing. In: *2010 First International Conference on Pervasive Computing, Signal Processing and Applications*. Harbin - China: [s.n.], 2010. p. 1039–1042. Citado na página 45.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, n. 6, p. 610–621, Nov 1973. Citado 2 vezes nas páginas 32 e 76.

HAYKIN, S. *Neural networks: a comprehensivefoundation*. 2th. ed. [S.l.]: Pearson Education, 2001. ISBN 0132733501. Citado 2 vezes nas páginas 38 e 78.

HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <http://arxiv.org/abs/1512.03385>. Citado 2 vezes nas páginas 33 e 76.

HECHT-NIELSEN, R. The meaning of synchronous distributed algorithms run on asynchronous distributed systems. In: *International 1989 Joint Conference on Neural Networks*. Washington - USA: [s.n.], 1989. p. 593–605 vol.1. Citado na página 38.

HTIKE, K. K.; KHALIFA, O. O. Comparison of supervised and unsupervised learning classifiers for human posture recognition. *Computer and Communication Engineering (ICCCE), 2010 International Conference on*, p. 1–6, May 2010. Citado na página 35.

HUANG, D. C.; HUNG, K. D. Leukocyte nucleus segmentation and recognition in color blood-smear images. In: *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. Graz - Austra: [s.n.], 2012. p. 171–176. Citado na página 48.

HUANG, D. C.; HUNG, K. D.; CHAN, Y. K. An adaptive leukocyte nucleus segmentation using genetic algorithm. In: *2012 International Symposium on Intelligent Signal Processing and Communications Systems*. New Taipei City - Taiwan: [s.n.], 2012. p. 559–563. Citado na página 45.

HUTTER, K. J.; STÖHR, M. Rapid detection of mutagen induced micronucleated erythrocytes by flow cytometry. *Histochemistry*, v. 75, n. 3, p. 353–362, Sep 1982. ISSN 1432-119X. Disponível em: <https://doi.org/10.1007/BF00496738>. Citado na página 21.

INTEL. *Color Models*. 2010. <https://software.intel.com/en-us/node/503873>. Acessado em 22/08/2017. Citado na página 26.

L FRIEDMAN J, O. R. S. C. B. *Classification and Regression Trees*. [S.l.]: Routledge, 1984. Citado na página 78.

LABATI, R. D.; PIURI, V.; SCOTTI, F. All-idb: The acute lymphoblastic leukemia image database for image processing. In: *2011 18th IEEE International Conference on Image Processing*. Brussels - Belgium: [s.n.], 2011. p. 2045–2048. Citado 4 vezes nas páginas 9, 59, 60 e 63.

LARSEN, R. T.; MARX, M. L. *An Introduction to Mathematical Statistics and Its Applications*. [S.l.]: Prentice Hall, 2000. Citado na página 77.

LE, D.-K. T. et al. An automated framework for counting lymphocytes from microscopic images. In: *2015 International Conference and Workshop on Computing and Communication (IEMCON)*. Vancouver - Canada: [s.n.], 2015. p. 1–6. Citado 3 vezes nas páginas 8, 48 e 49.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219. Citado 3 vezes nas páginas 8, 33 e 34.

LI, Y. et al. Segmentation of white blood cell from acute lymphoblastic leukemia images using dual-threshold method. *Computational and Mathematical Methods in Medicine*, v. 2016, p. 1–12, 01 2016. Citado 2 vezes nas páginas 8 e 47.

LIU, W. et al. Optimal color design of psychological counseling room by design of experiments and response surface methodology. *PloS one*, v. 9, p. e90646, 03 2014. Citado 2 vezes nas páginas 7 e 27.

LODDO, A. et al. A computer-aided system for differential count from peripheral blood cell images. In: *2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*. Naples - Italy: [s.n.], 2016. p. 112–118. Citado 3 vezes nas páginas 8, 50 e 88.

LUDWIG, O.; MONTGOMERY, E. *Redes Neurais: fundamentos e aplicações com programas em C*. [S.l.]: Editora Ciência Moderna, 2001. Citado na página 39.

LUGER, G. F. *Inteligência Artificial*. [S.l.]: Pearson, 2013. Citado na página 37.

MADHUKAR SOS AGAIAN, A. T. C. M. New decision support tool for acute lymphoblastic leukemia classification. *Proc.SPIE*, v. 8295, p. 8295 – 8295 – 12, February 2012. Citado na página 53.

MELO, M.; SILVEIRA, C. da. *Leucemias e Linfomas*. [S.l.]: Rubio, 2013. Citado na página 18.

MITCHELL, T. M. *Machine Learning*. [S.l.]: McGraw-Hill, 1997. Citado na página 35.

NEOH, S. C. et al. An intelligent decision support system for leukaemia diagnosis using microscopic blood images. *Scientific Reports*, v. 5, p. 1–14, October 2015. Citado 2 vezes nas páginas 55 e 56.

OJALA, T.; PIETIKAINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, v. 29, n. 1, p. 51–59, January 1996. Citado 3 vezes nas páginas 32, 33 e 76.

OLIPHANT, T. et al. *A guide to NumPy*. 2006. [Online; accessed jul/16/2018]. Disponível em: <http://www.numpy.org/>. Citado na página 40.

OLIVEIRA, L. de M. *Segmentação Fuzzy de Imagens e Vídeos*. Dissertação (Mestrado) — Pós-Graduação em Sistemas e Computação - Universidade Federal do Rio Grande do Norte, Rio Grande do Norte - RN, Fevereiro 2007. Citado na página 26.

OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62–66, Jan 1979. ISSN 0018-9472. Citado na página 24.

PATEL, N.; MISHRA, A. Automated leukaemia detection using microscopic images. In: *Second International Symposium on Computer Vision and the Internet (VisionNet'15)*. Kerala - India: [s.n.], 2015. p. 635–642. Citado na página 49.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 40 e 78.

PERINI, D. G. *Leucemia Linfoide Aguda - LLA*. 2016. <http://www.abrale.org.br/lla/o-que-e>. Acessado em 31/07/2017. Citado 2 vezes nas páginas 18 e 77.

PIURI, V.; SCOTTI, F. Morphological classification of blood leucocytes by microscope images. In: *2004 IEEE International Conference on Computational Intelligence for Measurements Systems and Applications, CIMSA*. Boston - USA: [s.n.], 2004. p. 103–108. Citado 4 vezes nas páginas 51, 52, 53 e 76.

PUTZU, L.; CAOCCI, G.; RUBERTO, C. D. Leucocyte classification for leukaemia detection using image processing techniques. *Artificial Intelligence in Medicine*, v. 62, p. 179–191, September 2014. Citado 4 vezes nas páginas 9, 54, 55 e 79.

RAMOSER, H. et al. Leukocyte segmentation and classification in blood-smear images. In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. Beijing - China: [s.n.], 2005. p. 3371–3374. Citado na página 52.

RAZAVIAN, A. S. et al. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014. Disponível em: <http://arxiv.org/abs/1403.6382>. Citado 2 vezes nas páginas 34 e 76.

RODRIGUES, L. F. et al. Leukocytes classification in microscopy images for acute lymphoblastic leukemia identification. In: *XII Workshop de visão computacional (WVC XII)*. Mato Grosso do Sul - Brasil: [s.n.], 2016. p. 1 –5. Citado na página 56.

ROSEBROCK, A. *Practical Python and OpenCV*. [S.l.]: PyImageSearch.com, 2016. Citado na página 24.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain [j]. *Psychol. Review*, v. 65, p. 386 – 408, 12 1958. Citado 2 vezes nas páginas 38 e 78.

SCOTTI, F. Automatic morphological analysis for acute leukemia identification in peripheral blood microscope images. In: *CIMSA. 2005 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2005.* Giardini-Naxos - Italy: [s.n.], 2005. p. 96–101. Citado na página 53.

SCOTTI, F. Robust segmentation and measurements techniques of white cells in blood microscope images. In: *2006 IEEE Instrumentation and Measurement Technology Conference Proceedings*. Sorrento - Italy: [s.n.], 2006. p. 307–316. Citado 4 vezes nas páginas 8, 19, 42 e 43.

SELVARAJ, S.; KANAKARAJ, M. Naive bayesian classifier for acute lymphocytic leukemia detection. *ARPN Journal of Engeneering and Applied Sciences*, v. 10, n. 16, p. 6888 – 6891, September 2015. Citado na página 56.

SILVA, M.; BOUZAS, L.; FIGUEIRA, A. Manifestações tegumentares da doença enxerto contra hospedeiro em pacientes transplantados de medula óssea. *An Bras Dermatol*, v. 80, n. 1, p. 801–823, October 2005. Citado na página 23.

SILVEIRA, G.; BULLOCK, B. *Machine Learning*. [S.l.]: Casa do Código, 2017. Citado 2 vezes nas páginas 32 e 35.

SOLOMON, C.; BRECKON, T. *Fundamentos de Processamento Digital de Imagens*. [S.l.]: LTC, 2013. Citado na página 28.

VOGADO, L. H. et al. Unsupervised leukemia cells segmentation based on multi-space color channels. In: *2016 IEEE International Symposium on Multimedia*. San Jose - California: [s.n.], 2016. p. 451–456. Citado na página 46.

VOGADO, L. H. S. et al. Um sistema de diagnostico de leucemia utilizando cnn's pre-treinadas e um comite de classificador. In: *3rd Escola Regional de Informatica do Piaui*. Picos – Piaui: [s.n.], 2017. p. 2–16. Citado na página 77.

VOGADO, L. H. S. et al. Um sistema de diagnóstico de leucemia utilizando cnn's pré-treinadas e um comitê de classificadores. In: *37º Workshop de Informática Médica (WIM 37)*. Sao Paulo - Brasil: [s.n.], 2017. p. 2020 –2029. Citado 2 vezes nas páginas 57 e 87.

WOELFEL, R. *Leukemia: From Diagnosis to Winning the Battle*. [S.l.]: Paperback, 2017. Citado na página 23.

WONG, S. C. et al. Understanding data augmentation for classification: When to warp? *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, p. 1–6, Nov 2016. Citado na página 34.

WU, Q. et al. Segmentation of leukocytes in blood smear images using color processing mechanism inspired by the visual system. In: *2009 2nd International Conference on Biomedical Engineering and Informatics*. Tianjin - China: [s.n.], 2009. p. 1–4. Citado 2 vezes nas páginas 8 e 44.

ZACK, G. W.; ROGERS, E. Automatic measurement of sister chromatid exchange frequency. *The journal of histochemistry and cytochemistry : official journal of the Histochemistry Society*, v. 25, p. 741–53, 08 1977. Citado na página 46.